

WRDC-TR-90-8007
Volume VIII
Part 9

AD-A248 919




INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 9 - Graph Definition Language Unit Test Plan

S. Barker

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209



September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

92-10246

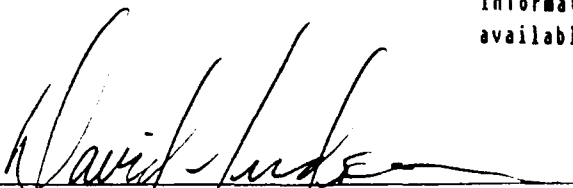

02 4 21 066

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

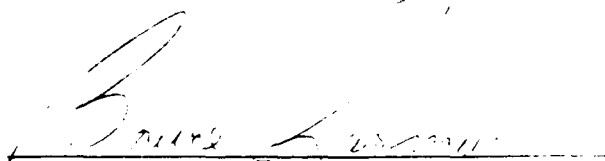
This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations


DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:


BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UTP620344210		5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8007 Vol. VIII, Part 9	
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services	6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI	
6c. ADDRESS (City, State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209		7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF	8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464	
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433-6533		10. SOURCE OF FUNDING NOS.	
11. TITLE See block 19		PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600
		TASK NO. F95600	WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Structural Dynamics Research Corporation: Wehrman, A.			
13a. TYPE OF REPORT Final Report	13b. TIME COVERED 4 / 1 / 87 - 12 / 31 / 90	14. DATE OF REPORT (Yr., Mo., Day) 1990 September 30	15. PAGE COUNT 133
16. SUPPLEMENTARY NOTES WRDC/MTI Project Priority 6203			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)	
FIELD	GROUP	SUB GR.	
1308	0905		
19. ABSTRACT (Continue on reverse if necessary and identify block number) This unit test plan establishes the methodology and procedures used to test the Graph Definition Language computer program. BLOCK 11: INTEGRATED INFORMATION SUPPORT SYSTEM Vol VIII - User Interface Subsystem Part 9 - Graph Definition Language Unit Test Plan			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson		22b. TELEPHONE NO. (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0423, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. J. P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

<u>SUBCONTRACTOR</u>	<u>ROLE</u>
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics	Responsible for User Interfaces, Research Corporation Virtual Terminal Interface, and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

	<u>Page</u>
SECTION 1.0 GENERAL	1-1
1.1 Purpose	1-1
1.2 Project References	1-1
1.3 Terms and Abbreviations	1-1
SECTION 2.0 DEVELOPMENT ACTIVITY	2-1
2.1 Statement of Pretest Activity	2-1
2.2 Pretest Activity Results	2-1
SECTION 3.0 SYSTEM DESCRIPTION	3-1
3.1 System Description	3-1
3.2 Testing Schedule	3-2
3.3 First Location Testing	3-3
3.4 Subsequent Location Testing	3-3
SECTION 4.0 SPECIFICATIONS AND EVALUATIONS	4-1
4.1 Test Specifications	4-1
4.2 Test Methods and Constraints	4-15
4.3 Test Progression	4-15
4.4 Test Evaluation	4-15
SECTION 5.0 TEST PROCEDURES	5-1
5.1 Test Description	5-1
5.2 Test Control	5-1
5.3 Test Procedures	5-1

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special



LIST OF ILLUSTRATIONS

<u>FIGURE</u>		<u>PAGE</u>
3-1	Interface Block Diagram	3-2
3-2	Application Environment in IISS	3-2
5-2	IISS Logon Screen	5-2
5-3	IISS Function Screen	5-3
5-4	Initial GRAFDE Screen	5-5
5-5	Test Data for GRAFDE	5-6
A-1	GRFTST1 Screen	A-2
A-2	GRFTST2 Screen	A-7
A-3	GRFTST3 Screen	A-10
A-4	GRFTST4 Screen	A-12
A-5	GRFTST5 Screen	A-14
A-6	GRFTST6 Screen	A-17
A-7	GRFTST7 Screen	A-20
A-8	GRFTST8 Screen	A-22
A-9	GRFTST9 Screen	A-24
A-10	GRFTST10 Screen	A-26
A-11	GRFTST11 Screen	A-29
A-12	GRFTST12 Screen	A-32
A-13	GRFTST13 Screen	A-35
A-14	GRFTST14 Screen	A-38
A-15	GRFTST15 Screen	A-41
A-16	GRFTST16 Screen	A-44
A-17	GRFTST17 Screen	A-47
A-18	GRFTST18 Screen	A-50
A-19	GRFTST19 Screen	A-53
A-20	GRFTST20 Screen	A-56
A-21	GRFTST21 Screen	A-59
A-22	GRFTST22 Screen	A-61
A-23	GRFTST23 Screen	A-63
A-24	GRFTST24 Screen	A-65
A-25	GRFTST25 Screen	A-67
A-26	GRFTST26 Screen	A-69
A-27	GRFTST27 Screen	A-72
B-1	GRAFFRM1 Screen	B-4
B-2	GRAFFRM2 Screen	B-6
B-3	GRAFFRM3 Screen	B-9
B-4	GRAFFRM4 Screen	B-12
B-5	GRAFFRM5 Screen	B-14
B-6	GRAFFRM6 Screen	B-17
C-1	RAP Function Screen	C-2

APPENDICES

A	SCREENS AND GDL FOR GRFTST	A-1
B	SCREENS AND GDL FOR GRAFDE	B-1
C	PRE-TEST PROCEDURES	C-1

SECTION 1

GENERAL

1.1 Purpose

This unit test plan establishes the methodology and procedures used to adequately test the capabilities of the computer programs identified collectively as the Graph Definition Language and known in this document as the GDL. The GDL is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Project References

- [1] Systran, ICAM Documentation Standards, IDS150120000C, 15 September 1983.
- [2] General Electric Company, System Design Specification, 7 February 1983.
- [3] Structural Dynamics Research Corporation, Form Processor Development Specification, DS 620344200, 31 March 1988.
- [4] Structural Dynamics Research Corporation, Forms Language Compiler Development Specification, DS 620244401A, 16 February 1987.
- [5] Structural Dynamics Research Corporation, Graph Definition Language Development Specification, DS 620344220, 15 December 1987.

1.3 Terms and Abbreviations

Application Generator (AG): A subset of the IISS User Interface that consists of software modules that generate IISS application code and associated form definitions based on a language input. The part of the AG that generates report programs is called the Report Writer. The part of the AG that generates interactive applications is called the Rapid Application Generator.

Application Interface (AI): A subset of the IISS User Interface that consists of the callable routines that are linked with applications that use the Form Processor or Virtual Terminal. The AI enables applications to be hosted on computers other than the host of the User Interface.

Application Process (AP): A cohesive unit of software that can be initiated as a unit to perform some function or functions.

Attribute: A field characteristic such as blinking, highlighted, black, etc., and various other combinations. Background attributes are defined for some forms or windows only. Foreground attributes are defined for items. Attributes may be permanent, i.e., they remain the same unless changed by the application program, or they may be temporary, i.e., they remain in effect until the window is redisplayed.

Closed Figure: A figure is closed if the path traced by a moving point returns to its starting position. The starting position may be arbitrarily assigned. "Fillarea" is synonymous with "closed figure".

Complex Figure: A figure is complex if the path traced by a moving point crosses itself. An arbitrary point may be determined to be contained within the traced boundary if a line drawn to infinity crosses the boundary an odd number of times. If the number of crossings is zero or even, the point is outside the traced boundary.

Dependent Data: Data correlated to a dependent variable.

Dependent Variable: A mathematical variable whose value is determined by that of one or more other variables in a function.

Device Drivers (DD): Software modules written to handle I/O for a specific kind of terminal. The modules map terminal-specific commands and data to a neutral format. Device Drivers are part of the UI Virtual Terminal.

Display List: An internal Form Processor list that contains only those forms that have been added to the screen and are currently displayed on the screen, along with information on where those forms are used.

Element: A graphics line or other primitive composed of graphics lines, such as an arc.

Field: In reference to the Forms Processor, "field" refers to any object on the open or display list. These objects can be forms, items, windows, etc.

In reference to graphs, "field" refers to a collection of one or more graph figures. A graph field can be an axis, curve, pie chart, grid, etc.

Figure: A collection of elements. A figure may be closed or open.

Fillarea: A collection of elements. A fillarea must be closed. "Closed figure" is synonymous with "fillarea".

Form: A structured view which may be imposed on windows or other forms. A form is composed of fields. These fields may be defined as forms, items, windows, prompts, non-graphics lines, and graphics.

Forms Definition Language (FDL): The language in which electronic forms are defined.

Forms Driven Form Editor (FD FE): A subset of the Form Editor which consists of a forms-driven application used to create and/or modify Form Definition files interactively.

Form Editor (FE): A subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor (FD FE) and the Forms Language Compiler (FLAN).

Form Hierarchy: A graphic representation of the way in which fields are related to their parent form.

Forms Language Compiler (FLAN): A subset of the Form Editor that consists of a batch process that accepts a series of Forms Definition Language (FDL) statements and produces form definition files as output.

Form Processor (FP): A subset of the IISS User Interface that consists of a set of callable execution-time routines available to an application program for form processing.

Graph: A picture correlated with data that alters as the data changes; by necessity, this is a dynamic (not pre-defined) picture. A graph may be imposed on windows or forms.

Graph Definition Language (GDL): An extension of the Forms Definition Language (FDL) which is used to define business graphs such as pie charts, X-Y plots, and bar charts.

Graph Figure: A collection of graphics primitives. The primitives can be circles, lines, arcs, etc.

Graphics Kernal System (GKS): A 2-dimensional graphics standard which is defined independently of any programming language.

Icon: A collection of figures and points that is pre-defined. An icon may be imposed on windows or forms. "Icon" is synonymous with "picture".

Independent Data: Data that is correlated to an independent variable.

Independent Variable: A mathematical variable whose value is specified first and determines the value of one or more other values in an expression or function. For example, in a business graph of sales versus month, month is the independent variable and sales is the dependent variable, because sales varies by month.

Integrated Information Support System (IISS): A test computing environment used to investigate, demonstrate, and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network (LAN).

Item: A non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined area where user data may be input/output.

Local Area Network (LAN): A privately owned network that offers reliable, high-speed communications channels optimized for connecting information processing equipment in a limited geographic area.

Message: Descriptive text which may be returned in the standard message line on the terminal screen. They are used to warn of errors or to provide other user information.

Message Line: A line on the terminal screen that is used to display messages.

Open Figure: A figure is open if the path traced by a moving point does not return to its starting position. The starting position may be arbitrarily assigned. "Polyline" is synonymous with "open figure".

Open List: An internal Form Processor list that contains all forms that the application has opened for use along with information on where the form is used.

Operating System (OS): Software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Page: An instance of a form in a window that is created whenever a form is added to a window.

Physical Device: A hardware terminal.

Picture: A collection of figures and points that is pre-defined. A picture may be imposed on a window or a form. "Picture" is synonymous with "icon".

Picture Definition Language (PDL): An extension of the Forms Definition Language (FDL) which allows the definition of any graphics picture.

Point: A marker or a symbol.

Polyline: A collection of elements. A polyline must be an open figure. "Open figure" is synonymous with "polyline".

Primitive: The smallest unit of graphic detail. A graphic primitive can be a line, point, arc, etc.

Qualified Name: The name of a field preceded by the hierarchy path so that it is uniquely identified.

Report Writer (RW): Part of the Application Generator (AG) that generates source code for report programs based on a language input.

Subform: A form that is used within another form.

Text Editor (TE): A subset of the IISS User Interface that consists of a file editor that is based on the text editing functions built into the Form Processor (FP).

User Data: Data which is either input by the user or output by the application programs to items.

User Interface (UI): A subsystem of IISS that controls the user's terminal and interfaces with the rest of the subsystem. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System (UIDS): A collection of IISS User Interface subsystems that is used by application programmers as they develop IISS applications. The UIDS includes the Form Editor (FE) and the Application Generator (AG).

User Interface Management System (UIMS): The run-time UI. It consists of the Form Processor (FP), Virtual Terminal (VT), Application Interface (AI), the User Interface Services (UIS), and the Text Editor (TE).

User Interface Services (UIS): A subset of the IISS User Interface that consists of a package of routines that aids users in controlling their environment. It includes message management, change password, and application definition services.

User Interface/Virtual Terminal Interface (UI/VTI): Another name for the User Interface.

Window: A dynamic area of a terminal screen on which pre-defined forms may be placed at run-time.

Window Manager: A facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor (FP).

SECTION 2

DEVELOPMENT ACTIVITY

2.1 Statement of Pretest Activity

During system development, the computer programs were tested progressively. Functionality was incrementally tested and, as bugs were discovered by this testing, the software was corrected.

Each form used in the UTP for business graphs was tested individually. This testing was conducted by the individual program developer in a manual mode. The developer manually entered data onto the screen and observed the results. Any errors were noted by the developer and corrections to the program were then made after the testing session.

2.2 Pretest Activity Results

Testing of the graph forms used in the UTP uncovered a few minor bugs which were then corrected and retested successfully. Testing included exceptional conditions and error conditions for data entered on the forms.

SECTION 3

SYSTEM DESCRIPTION

3.1 System Description

The GDL does not interface directly with end users as an application, rather the Graph Definition Language (GDL) is an extension to the Forms Definition Language (FDL) which enables the definition of graphs through the User Interface software. Physical terminals are assumed to have both video display and graphics, a textual keyboard, four cursor positioning keys or key sequences, a help key or key sequence, a message key, an entry key, and a quit key. The GDL must interface with the following software tools: the Forms Processor (FP), the Forms Compiler (FLAN), C language runtime routines, and the Application Generator (AG). They are used to create or modify FDL files and to create new FD files and finally to display these FD files at run time.

This section describes the interfaces within the graphics software. In the following figures, GI stands for graphics interface, GKS is the graphics software which builds the internal data structures, GKS2 is the actual graphics software, and AI is the application interface. The AI routines are the calls to the UI which specify the actions to be performed. The FP, GI, GKS, and GKS2 systems perform the required actions. For graphics, clipping is performed in the VTI. Existing graphics applications written using the Fortran GKS binding will be able to call the comparable routines in the GKS subsystem. It should be noted that if the Graph Definition Language is used, the application will not normally be using the Graph or Picture AI routines.

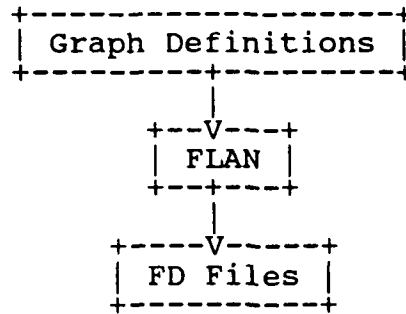


Figure 3-1 Compilation of Graph Definition

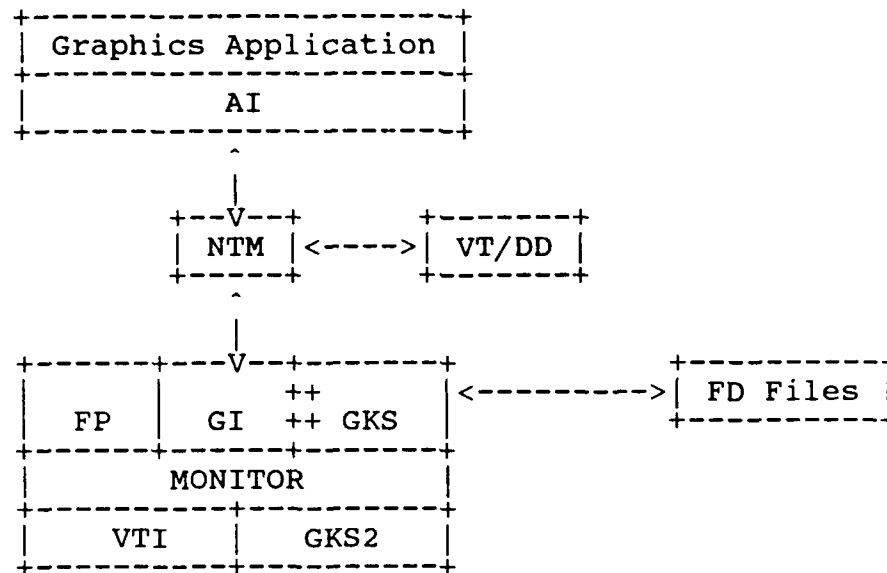


Figure 3-2 Application Environment in IISS

3.2 Testing Schedule

The production of business graphs is dependent upon the NTM subsystem of IISS, so testing of the graph capability of the Form Processor should be performed after the NTM has been successfully tested. Within the UI subsystem, this capability is part of the FP, VT, and FLAN, and therefore, there are no other restrictions.

3.3 First Location Testing

These tests of the business graphs capability of the Form Processor require the following:

Equipment: IISS VAX, Tektonix model 4100 or 4200 series terminal supported by the VT as listed in the UI Terminal Operator Guide. The terminal should have certain characteristics set. The tester should press the setup key and type the following:

```
code ansi
dal 30
bypass 00
flagging in/out
eol /~<CR>/
```

The above may be saved in nonvolatile memory by typing NSAVE. To exit setup mode, press the setup key again .

Support Software: The Integrated Information Support System; C run-time libraries.

Personnel: One integrator familiar with the UIMS.

Training: FP manuals have been previously provided with the past release.

Deliverables: The GDL subsystem of the UI.

Test Materials: This test uses the test programs GRFTST and GRAFDE and the forms defined in GRFTEST.FDL. Appendix C explains how to create the executables for these programs

Security Considerations: None.

3.4 Subsequent Location Testing

The requirements as listed above need to be met. Since the test consists of a routine that issues the Form Processor calls necessary to supply the data and display the graphical forms, there will be no differences in how testing should occur unless the tester wishes to script. Scripting is discussed in Section 5.

SECTION 4

TEST SPECIFICATIONS AND EVALUATIONS

4.1 Test Specification

The following functionality of the GDL is demonstrated by the test outlined in section 5:

List of Functions

GRAPH DEFINITION

1. bar
2. pie
3. line
4. independent axis
5. independent data

ATTRIBUTE DEFINITION

6. color
7. font
8. size
9. upvector
10. line width
11. line type
12. symbol
13. symbol frequency

DATA LOCATION

14. constant list
15. path list

CURVE DEFINITION

16. absolute display
17. additive display
18. dependent axis
19. independent data
20. shading
21. monochromatic shading
22. display
23. monochromatic display
24. legend label

LEGEND

- 25. enclosed
- 26. not enclosed
- 27. horizontal
- 28. vertical

PIE SEGMENT

- 29. explosion
- 30. shading
- 31. monochromatic shading
- 32. legend label
- 33. label
- 34. inside percent label
- 35. outside percent label
- 36. inside quantity label
- 37. outside quantity label

AXIS DEFINITION

- 38. length
- 39. log scale
- 40. linear scale
- 41. grid lines
- 42. fine grid lines
- 43. horizontal
- 44. vertical
- 45. location
- 46. label
- 47. maximum limit
- 48. minimum limit
- 49. minor tick marks
- 50. major tick marks by step
- 51. major tick marks by number
- 52. major tick mark labels

AUTOMATIC GENERATION

- 53. independent axis
- 54. dependent axis
- 55. tick marks
- 56. axis length
- 57. minimum axis value
- 58. maximum axis value
- 59. tick mark labels
- 60. legend labels
- 61. pie segments
- 62. pie segment percent label
- 63. automatic layout

Dynamic Creation and Alteration of Graph Definitions

GENERAL GRAPH DEFINITION

- 64. define a graph
- 65. define a graph location
- 66. delete graph
- 67. add graph label
- 68. remove graph labels
- 69. add where data is located clause
- 70. add a constant list of data
- 71. define an attribute bundle
- 72. delete an attribute bundle
- 73. define a graphics clipping window
- 74. define the graph extent within the window
- 75. add an independent axis
- 76. legend label
- 77. delete where data is located clause

LEGEND DEFINITION

- 78. add legend
- 79. delete legend

PIE DEFINITION

- 80. pie segment definition
- 81. delete pie segment labels
- 82. percent or quantity label to segment
- 83. pie segment label
- 84. delete pie segment

CURVE DEFINITION

- 85. define a curve
- 86. delete a curve

AXIS DEFINITION

- 87. delete tick mark labels
- 88. delete axis labels
- 89. delete axis definition
- 90. define maximum and minimum values on axis
- 91. location of axis
- 92. add tick mark labels
- 93. add axis labels
- 94. define an axis
- 95. add tick marks

- 96. polyline clipping
- 97. fillarea clipping
- 98. text clipping

Tables 4-1 and 4-2 show the direct correspondence between the test graphs and the functional requirements as listed in this section. These functions directly correspond to the detailed functional requirements of the Graph Definition Language Development Specification. The '.' indicates the tests for the functionality implemented in the current release. The '*' indicates functionality not yet implemented.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

Continued on next page

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
23																										
24					
25																		
26											
27																		
28	
29		
30		
31																										
32																						
33																						.		.	.	
34																						.		.	.	
35																						
36																						.		.	.	
37																						
38
39																										
40
41																										
42																										
43
44

Continued on next page

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

Continued on next page

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
67																										
68																										
69																										
70																										
71																										
72																										
73																										
74																										
75																										
76																										
77																										
78																										
79																										
80																										
81																										
82																										
83																										
84																										
85																										
86																										
87																										
88																										

Continued on next page

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
89																										
90																										
91																										
92																										
93																										
94																										
95																										
96	
97											
98	

Table 4-1 Matrix Mapping GDL Functions to Test Graphs

	AA	BB	CC	DD	EE	FF	GG	
1	.		.			.		
2		.			.			
3				.			.	
4	
5	
6	
7								*
8								*
9								*
10								*
11								*
12								*
13								*
14	
15	
16	
17	.		.				.	
18	
19								*
20	
21								*
22	.			.			.	

Continued on next page

	AA	BB	CC	DD	EE	FF	GG	
23								*
24			
25								
26			
27								
28			
29		.			.			
30		.			.			
31								
32								
33					.			
34					.			
35								
36								
37								
38	
39								*
40	
41	
42	.							
43	
44	

Continued on next page

	AA	BB	CC	DD	EE	FF	GG	
45	
46			
47	.							
48			
49	.							
50								
51	
52	
53								*
54								*
55	
56								*
57	
58	
59	
60	.							
61								*
62								
63								*
64								*
65								*
66								*

Continued on next page

	AA	BB	CC	DD	EE	FF	GG	
67								*
68								*
69								*
70								*
71								*
72								*
73								*
74								*
75								*
76								*
77								*
78								*
79								*
80								*
81								*
82								*
83								*
84								*
85								*
86								*
87								*
88								*

Continued on next page

	AA	BB	CC	DD	EE	FF	GG	
89								*
90								*
91								*
92								*
93								*
94								*
95								*
96	.							
97	.							
98								

Table 4-2 Matrix Mapping GDL Functions to Test Graphs

The test activities labeled A through GG map to the figures in Appendices A and B as follows:

- A - Figure A-1
- B - Figure A-2
- C - Figure A-3
- D - Figure A-4
- E - Figure A-5
- F - Figure A-6
- G - Figure A-7
- H - Figure A-8
- I - Figure A-9
- J - Figure A-10
- K - Figure A-11
- L - Figure A-12
- M - Figure A-13
- N - Figure A-14
- O - Figure A-15
- P - Figure A-16
- Q - Figure A-17
- R - Figure A-18
- S - Figure A-19

T - Figure A-20
U - Figure A-21
V - Figure A-22
W - Figure A-23
X - Figure A-24
Y - Figure A-25
Z - Figure A-26
AA - Figure A-27
BB - Figure B-1
CC - Figure B-2
DD - Figure B-3
EE - Figure B-4
FF - Figure B-5
GG - Figure B-6

4.2 Testing Methods and Constraints

The testing as outlined in Section 5 must be followed. The required input is stated for each test. This testing tests the normal mode of operation of these functions and does not completely exercise all the error combinations that a user of GDL might create by faulty definitions of the graph. These tests have been done, however, through the normal testing done by the developer of these functions. IISULIB and IISSSLIB should point to the default directory. No additional constraints are placed on this unit test besides those listed in sections 3.2 and 3.3 of this unit test plan.

4.3 Test Progression

The progression of testing of the GDL is fully outlined in Section 5 of this unit test plan. This progression should be followed exactly to insure the successful testing of this IISS configuration item.

4.4 Test Evaluation

The test results are evaluated by comparing the information returned on the various output screens with that specified as successful for the given test. As outlined in section 5, each test of GDL functionality provides a screen with the output for a successful test. The data necessary for input is done automatically before the output screen. The only differences found should be the date and time stamps on the IISS Function Screen (Figure 5-3) and the first test output screen (Figure A-1).

SECTION 5

TEST SPECIFICATIONS AND EVALUATIONS

5.1 Test Description

Two test programs are used to test the GDL. The test program GRFTST uses explicit FP calls to place the form within a window, place data within the form fields where the graph data is to be located, display the graph form, and remove the graph form from the window. Since the program issues the data using data from internal arrays, no data entry is required by the tester.

The test program GRAFDE is an interactive application that is generated using the Rapid Application Generator. GRAFDE represents user entered data as a pie, bar, or line graph.

5.2 Test Control

As outlined, this unit test is a manual test which may be done by anyone. The required input data for each function being tested, the resulting successful output and the order of the testing are completely specified below. Accurate observation of the resulting successful output must be made to ensure the unit test was done properly.

5.3 Test Procedures

To run the unit test plan in the VAX/VMS environment as outlined below, one must be logged onto an IISS account. The NTM must be up and running and the UI logical names IISSFLIB, IISSULIB, IISSSLIB, and IISSMLIB must be set properly at the group level. IISSFLIB points to the directory containing system form definitions (FD files). IISSULIB points to the directory containing the user's form definitions (FD files). IISSSLIB points to the directory containing the user's form definition source files (FDL files). IISSMLIB points to the directory containing the UI error and help messages (MSG files). To perform this test IISSULIB and IISSSLIB must be pointing to the default directory.

Assuming the NTM is up and running, an IISS user may start this test as follows:

```
$ SET DEF <to directory containing NTM environment>
$ TEK4100
```

These commands start up the TEK4100 device driver.

5.3.1 Access to GDL Test Programs

Following entry of the system command "TEK4100" which activates the User Interface the following form appears:

```

USER ID: _____
PASSWORD: _____
ROLE: _____

```

Figure 5-1 IISS Logon Screen

- (1) USER ID is the identification name of the user, and is 1 to 10 alpha-numeric characters. USER ID is input as "MORENC".
- (2) PASSWORD must be the password associated with the USER ID, and is 1 to 10 alpha-numeric characters. PASSWORD was input as "STANLEY".

- (3) ROLE is any of the identifiers which are associated with the USER ID, and is 1 to 10 alpha-numeric characters. It will be checked against functions and applications which are selected by the user. ROLE is input as "MANAGER".

When this form is correctly completed and the <ENTER> key is pressed, the IISS Function Screen is displayed.

I I S S T E S T B E D V E R S I O N 2.3			
DATE: __/__/__	TIME__:__:__	USER ID: _____	ROLE: _____
FUNCTION: _____		DEVICE TYPE: _____	DEVICE NAME: _____
Msg: 0		application	

Figure 5-2 IISS Function Screen

When this form appears, the cursor is located in the input field labeled FUNCTION. The items in the form are summarized below:

- (1) DATE contains the current date. This may not be changed by the user.
- (2) TIME contains the current time. This may not be changed by the user.

- (3) USER ID is the user's identification that was entered in the previous form. This may not be changed by the user.
- (4) ROLE is the currently active role and was entered in the previous form. This may be changed at any time.
- (5) FUNCTION is the function the user desires to activate.

To run the GDL test programs, proceed as described in the following sections.

5.3.2 Running the GRFTST Program

To run the GRFTST program, enter "GRFTST" in the FUNCTION field on the IISS Function Screen and press the <ENTER> key. This program produces the 27 graphs shown in Appendix A. Test Graph A is displayed when the program begins. Each succeeding graph is displayed by repeatedly pressing the <ENTER> key. Before proceeding to the next graph, the graph displayed on the terminal screen should be compared with the corresponding graph in Appendix A. When all 27 graphs have been displayed and compared, a final press of the <ENTER> key terminates the program and redisplay the IISS Function Screen.

5.3.3 Running the GRAFDE Program

To run the GRAFDE program, enter "GRAFDE" in the FUNCTION field on the IISS Function Screen and press the <ENTER> key. The following screen is displayed.

ENGINEERING CHANGE ANALYSIS CENTER				DATE: 12/10/87
CHANGE COST GRAPH				TIME: 11:46:50
	Cur.Yr.	Cur.Yr.-1	Cur.Yr.-2	Cur.Yr.-3
Rework:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Scrap :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Misc :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Enter data and press <PF5> - Pie , <PF6> - Bar, <PF7> - Line **				
MSG: <input type="text"/> 0				
application				

Figure 5-3 Initial GRAFDE Screen

Enter the data as shown in Figure 5-4 and press the appropriate function key to produce the desired graph as described in Table 5-1.

ENGINEERING CHANGE ANALYSTS CENTER					DATE: 12/21/87
CHANGE COST GRAPH					TIME: 8:22:15
	Cur. Yr.	Cur. Yr. -1	Cur. Yr. -2	Cur. Yr. -3	
Rework:	20	10	25	30	
Scrap :	40	35	30	10	
Misc :	50	30	20	10	
Enter data and press <PF5> - Pie , <PF6> - Bar, <PF7> - Line					
MSG: 0					application

Figure 5-4 Test Data for GRAFDE

A total of six separate screens may be presented using the data. To display the appropriate graph, press the indicated function key.

PFKEY	APPENDIX FIGURE	DESCRIPTION
5	B-1	Pie chart with percentages outside
6	B-2	Horizontal bar chart
7	B-3	Line graph
9	B-4	Pie chart with percentages inside and labels
10	B-5	Vertical bar chart
11	B-6	Line chart with area under curves shaded
4		Quit application

Table 5-1 PFKEY and Figure Correlation

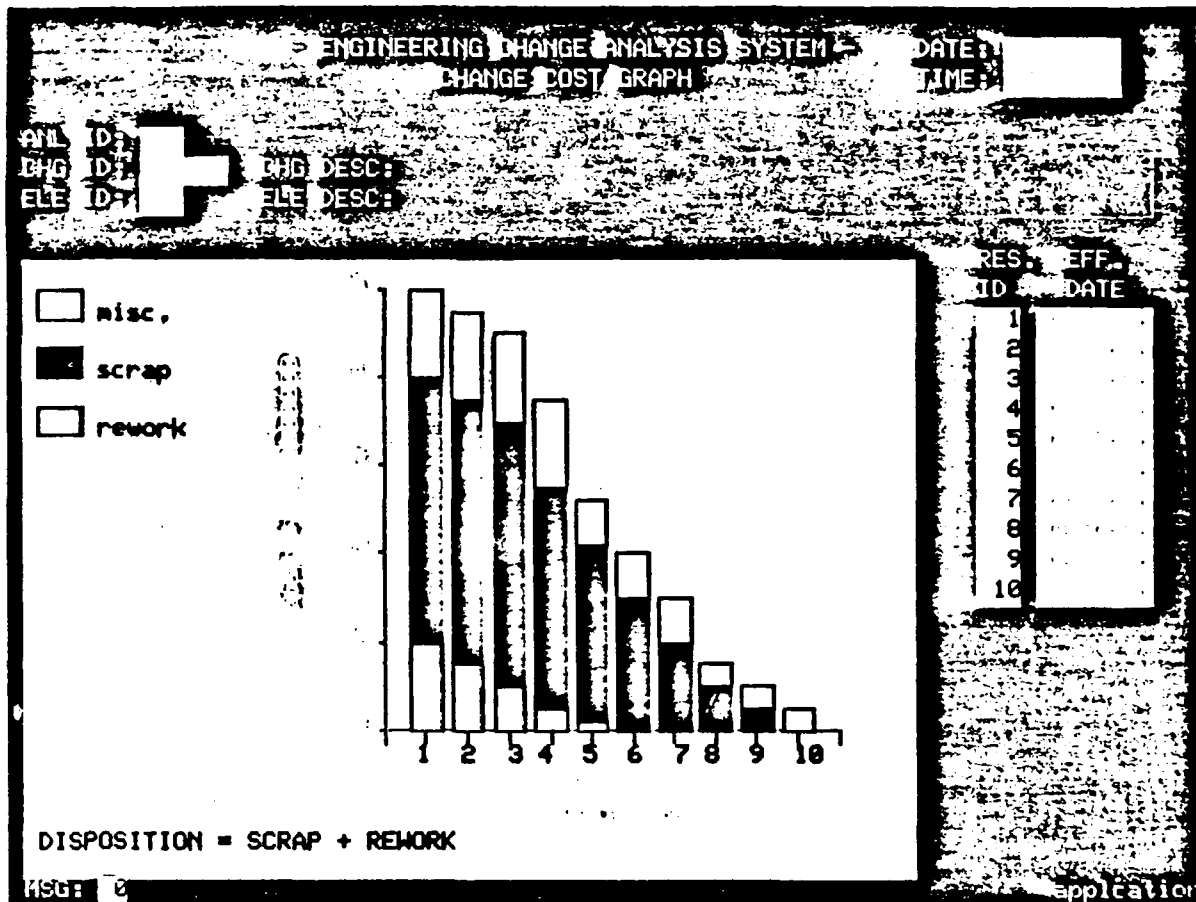
The screens displayed should be compared with the indicated graph figures in Appendix B. Only the time/date stamps should differ.

APPENDIX A

SCREENS AND GDL FOR GRFTST

This appendix contains all the screens for the first test of the Graph Definition Language. The necessary FDL follows the screens.

Figure A-1 and corresponding GDL



```
create form grftst1
  size 80 by 30
  prompt at 2 20 "- ENGINEERING CHANGE ANALYSIS SYSTEM -"
  prompt at 3 30 "CHANGE COST GRAPH"
  attribute out (background white, display blue, guarded,
    nowrite)
  attribute out2 (background blue, display yellow, guarded)
  attribute out3 (background red, display black, guarded)
  attribute nosee (guarded, hidden)

item curdat
  at 2 68
  size 8
  prompt at 2 62 "DATE:"
  value '._date'
  display as out

item curtim
  at 3 68
  size 8
  prompt at 3 62 "TIME:"
  value '._time'
  display as out

item anlid
  at 5 10
  size 3
  prompt at 5 2 "ANL ID:"
  display as out2

item chgid
  at 6 10
  size 6
  prompt at 6 2 "CHG ID:"
  display as out2

item chgdsc
  at 6 28
  size 50
  prompt at 6 18 "CHG DESC:"
  display as out3

item eleid
  at 7 10
  size 3
  prompt at 7 2 "ELE ID:"
  display as out2
```



```
item eledsc
  at 7 28
  size 50
  prompt at 7 18 "ELE DESC:"
  display as out3

graph cstgrf
  at 9 2
  display as blue
  size 60 by 21

form csttab
  at 9 65
  display as black
  size 16 by 21

create bar graph cstgrf
  using ('csttab.ids' axis ax1)
  attribute a line (display yellow)
  attribute b prompt (display white)
  attribute c prompt (display green)
  attribute d prompt (display red)
  legend at 2 2
  label display as d, at 20 2 "DISPOSITION = SCRAP + REWORK"

curve rework
  'csttab.rewcst' using axis ax2
  legend c "rework"
  absolute

curve misc
  'csttab.msccst'
  additive using curve scrap
  legend c "misc."

curve scrap
  'csttab.scrbst'
  additive using curve rework
  legend c "scrap"
```

```
axis ax1
  horizontal
  display as a
  at 16 25
  min 0
  size 30
  label b "          result id"
  tick every 1 d " " "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"

axis ax2
  at 16 25
  size 15
  label b "          cost $ "
  vertical
  min 0
  display as a

create form csttab
  prompt at 1 2 "RES."
  prompt at 2 2 "ID"
  prompt at 1 8 "EFF."
  prompt at 2 8 "DATE"
  attribute hid (hidden, guarded)

item dates (10 v 0)
  size 8
  at 3 6
  display as magenta

item ids (10 v 0)
  size 3
  at 3 2
  domain (numeric)
  display as cyan

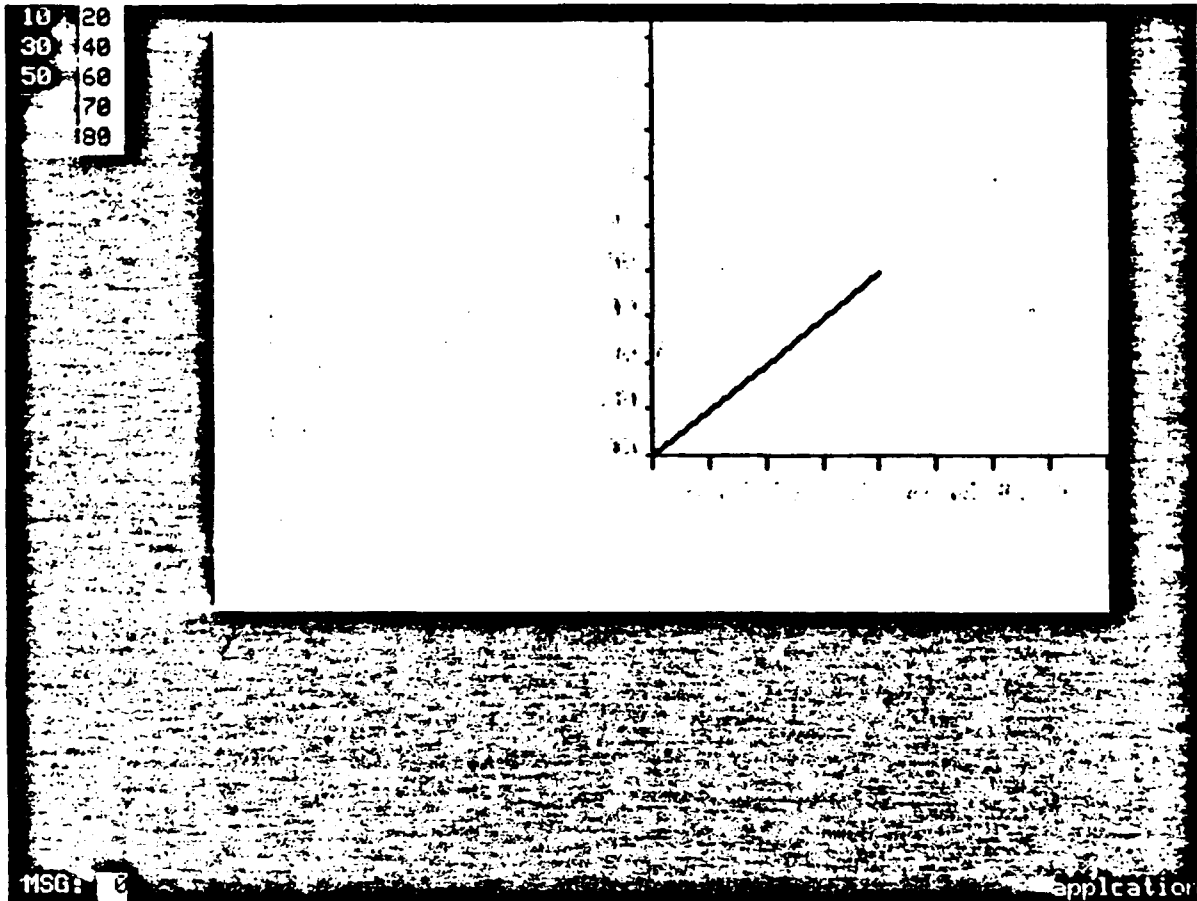
item mscst (10 v 0)
  size 6
  at 3 15
  display as hid
  domain (numeric)

item srcst (9 v 0)
  at 3 35
  domain (numeric)
  display as hid
  size 6
```

UTP620344210
30 September 1990

item rewcst (5 v 0)
at 3 45
size 6
domain (numeric)
display as hid

Figure A-2 and corresponding GDL



```
create form grftst2
  size 80 by 30
```

```
item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)
```

```
item i2 (5 v 0)
  display as yellow
  at 1 6
  size 3
  domain (numeric)
```

```
graph grf1
  at 1 15
  size 60 by 20
  display as blue
```

```
create line graph grf1
  using (1, 2, 3, 4, 5 axis ax1)
  attribute a fill (display cyan)
  attribute b line (display magenta)
  attribute xy prompt (display yellow)
  attribute x line (display yellow)
  attribute c prompt (display white)
  attribute d line (display green)
  background blue
```

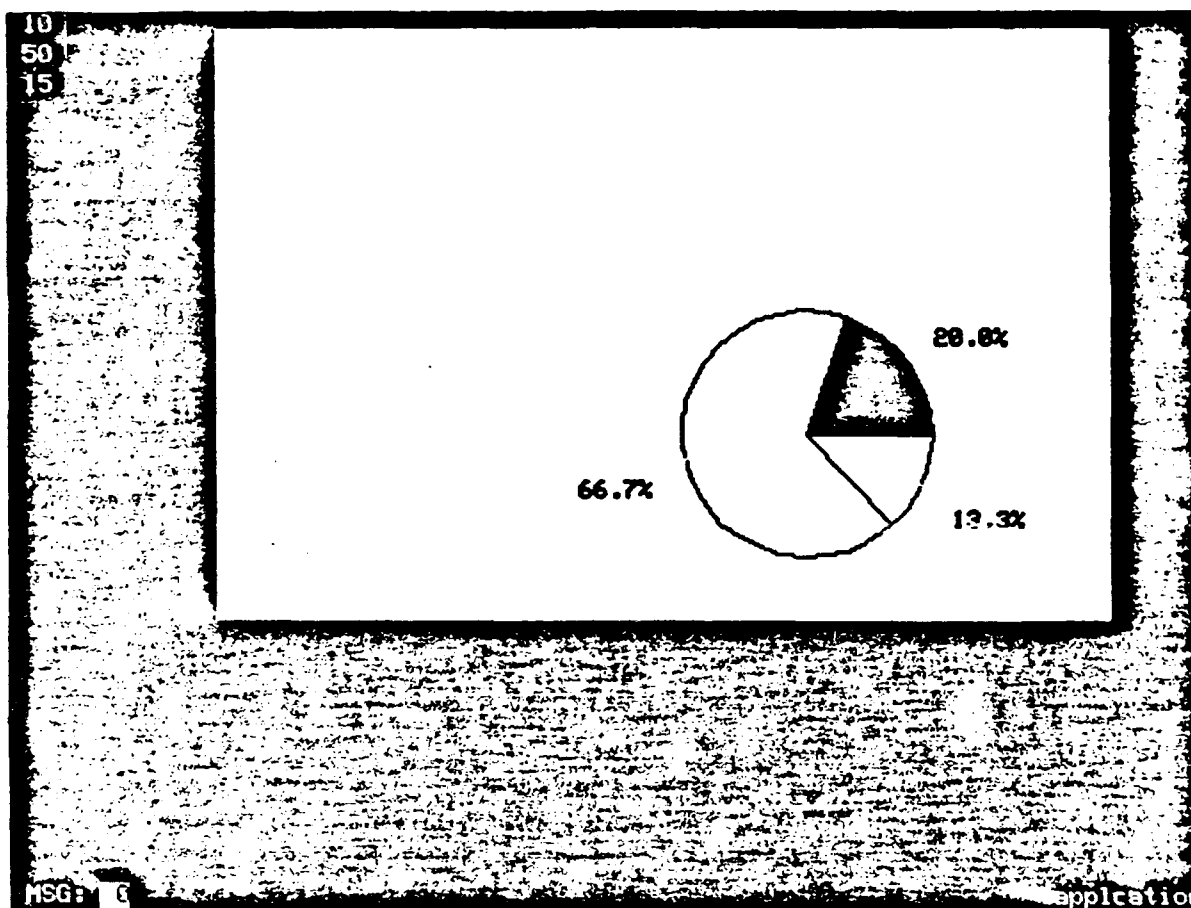
```
curve aaa
  'grftst2.i1' using axis ax2
  absolute
```

```
curve two
  'grftst2.i2'
  additive using curve aaa
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
```

axis ax2
at 15 30
size 15
vertical
display as x

Figure A-3 and corresponding GDL



```
create form grftst3
    size 80 by 30

form fgrf (3 v 0)
    at 1 1
    size 5 by 1

graph grf2
    at 1 15
    size 60 by 20
    display as blue

create pie graph grf2
    at 10 30
    size 20 by 8
    using ('grftst3.fgrf(*) .il')

pie 1
    shade color red

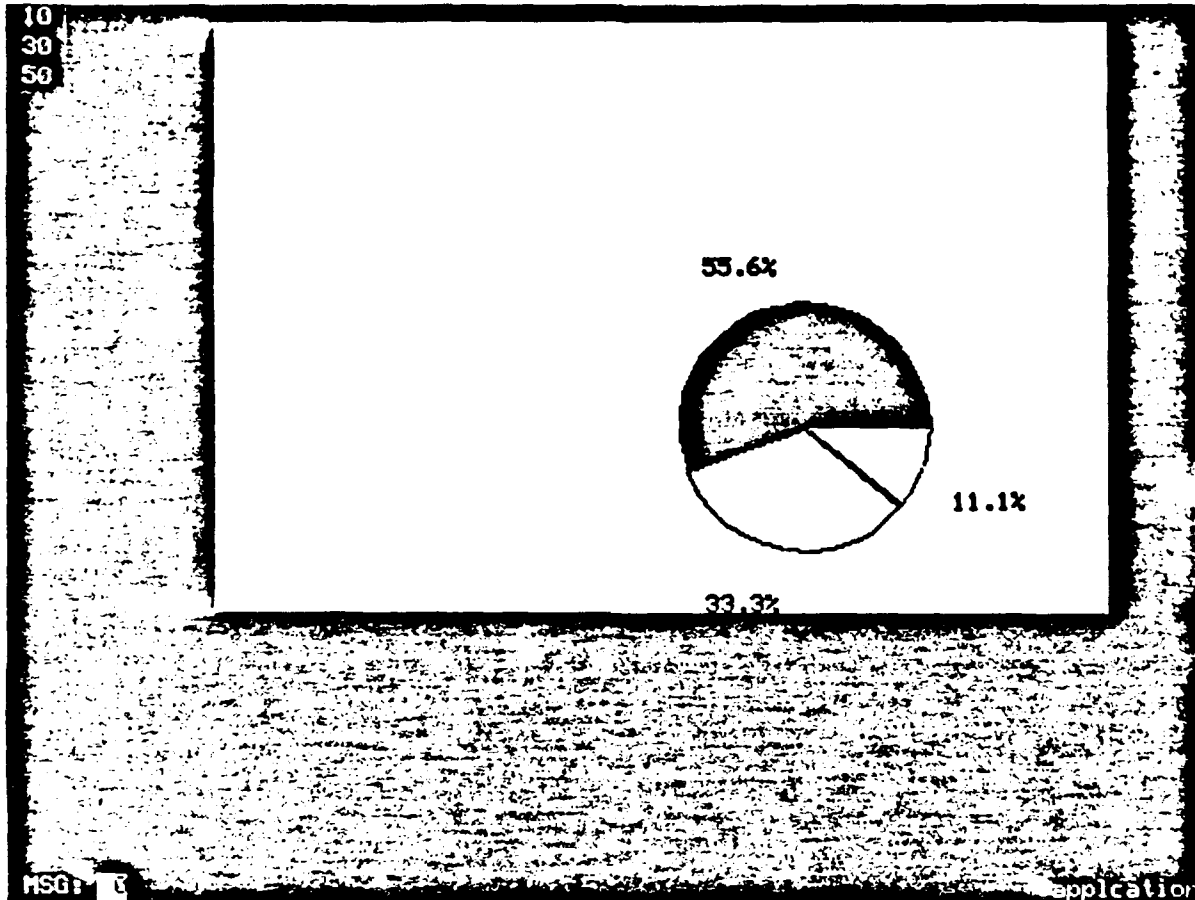
pie 2
    shade color magenta

pie 3
    shade color white

create form fgrf

item il
    display as red
    at 1 2
    size 3
    domain (numeric)
```


Figure A-4 and corresponding GDL



```
create form grftst4
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

graph grf3
    at 1 15
    size 60 by 20
    display as blue

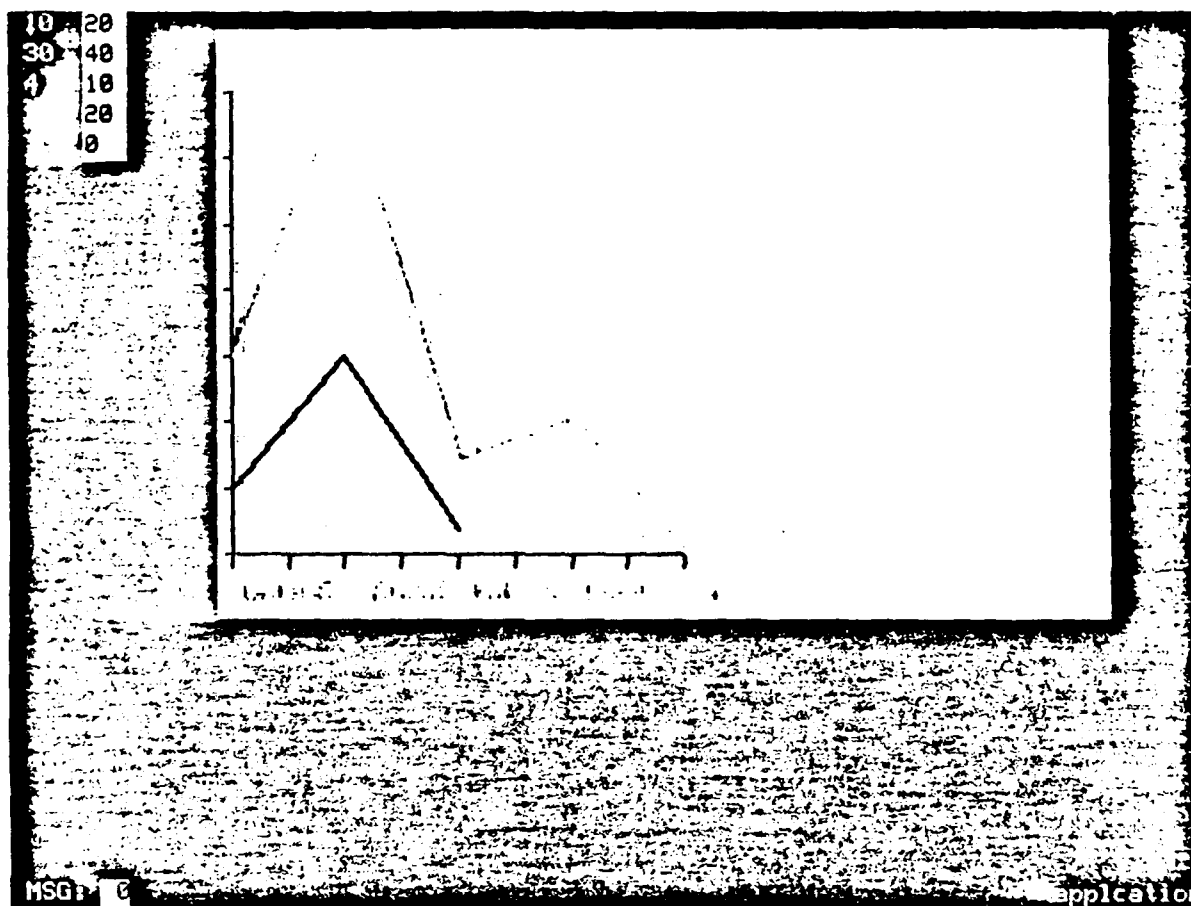
create pie graph grf3
    at 10 30
    size 20 by 8
    using ('grftst4.i1')

pie 1
    shade color red

pie 2
    shade color magenta
    explode 2

pie 3
    shade color white
```

Figure A-5 and corresponding GDL



```
create form grftst5
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf4
    at 1 15
    size 60 by 20
    display as blue

create line graph grf4
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    background blue

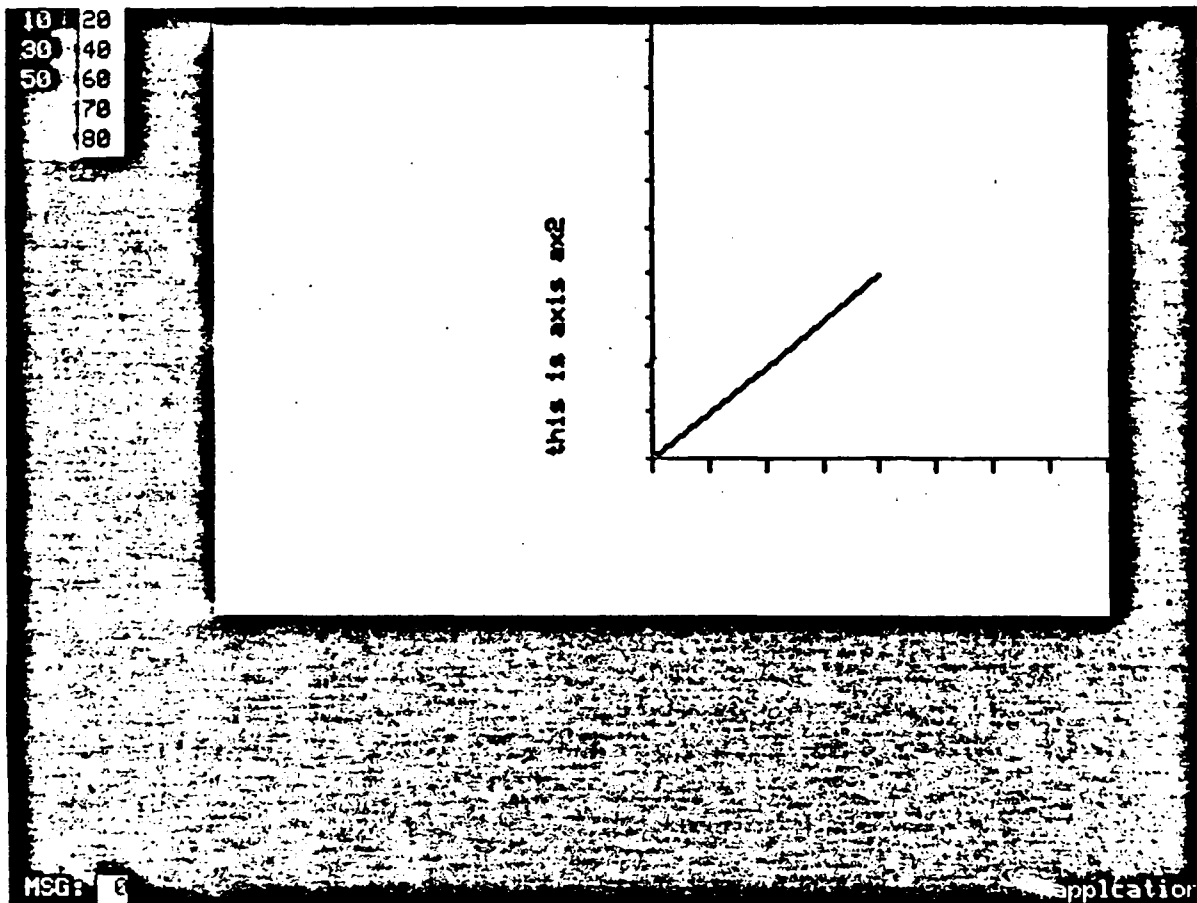
curve aaa
    'grftst5.i1' using axis ax2
    absolute

curve two
    'grftst5.i2'
    additive using curve aaa

axis ax1
    horizontal
    display as x
    at 18 2
    size 30
    label c "this is a label"
```

```
axis ax2
  at 18 2
  size 15
  label c "this is a label"
  vertical
  display as x
```

Figure A-6 and corresponding GDL



```
create form grftst6
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

item i2 (5 v 0)
  display as yellow
  at 1 6
  size 3
  domain (numeric)

graph grf5
  at 1 15
  size 60 by 20
  display as blue

create line graph grf5
  using (1, 2, 3, 4, 5 axis ax1)
  attribute a fill (display cyan)
  attribute b line (display magenta)
  attribute xy prompt (display yellow)
  attribute x line (display yellow)
  attribute c prompt (display white)
  attribute d line (display green)
  background blue

curve aaa
  'grftst6.i1' using axis ax2
  absolute

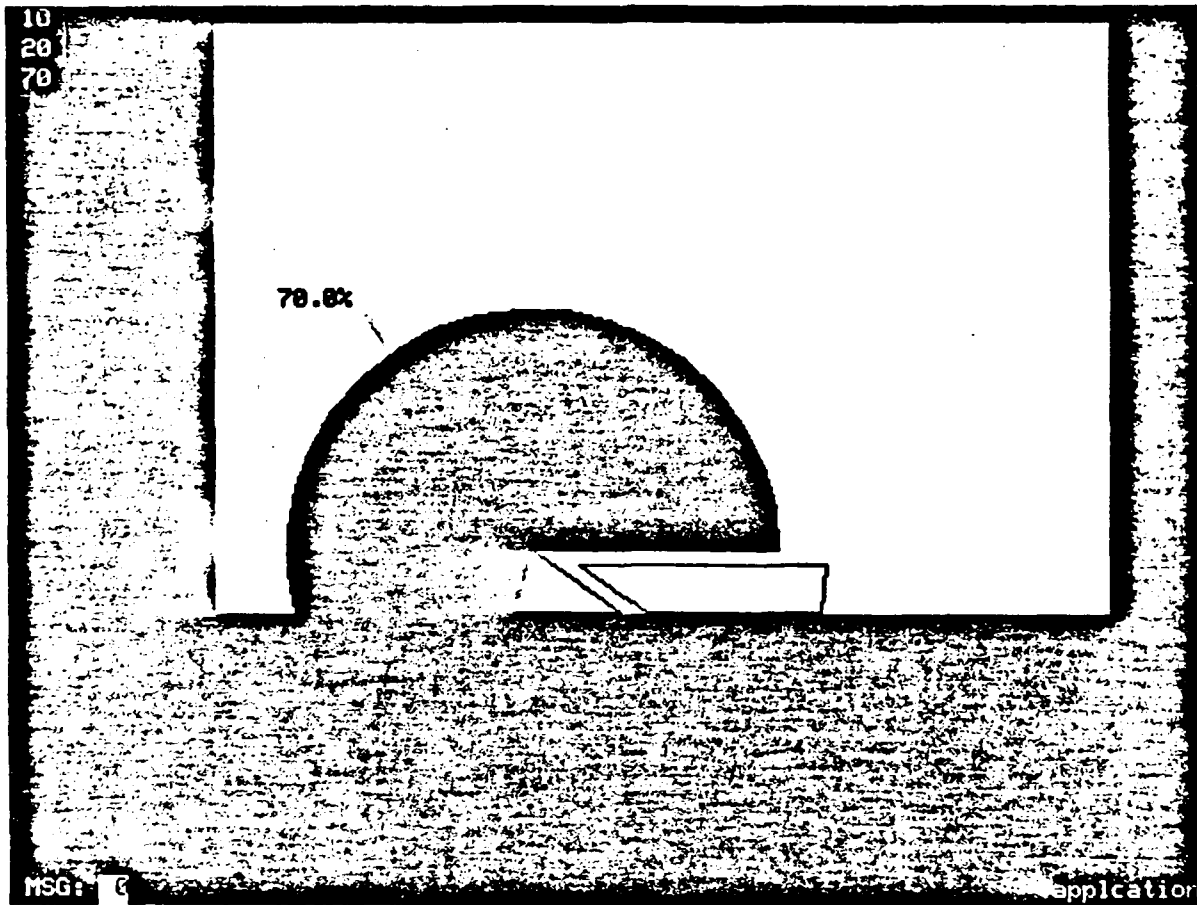
curve two
  'grftst6.i2'
  additive using curve aaa

axis ax1
  horizontal
  display as x
  label c "this is axis ax1"
  at 15 30
  size 30
```

UTP620344210
30 September 1990

```
axis ax2
  at 15 30
  size 15
  vertical
  display as x
  label xy "this is axis ax2"
```


Figure A-7 and corresponding GDL



```
create form grftst7
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

graph grf6
    at 1 15
    size 60 by 20
    display as blue

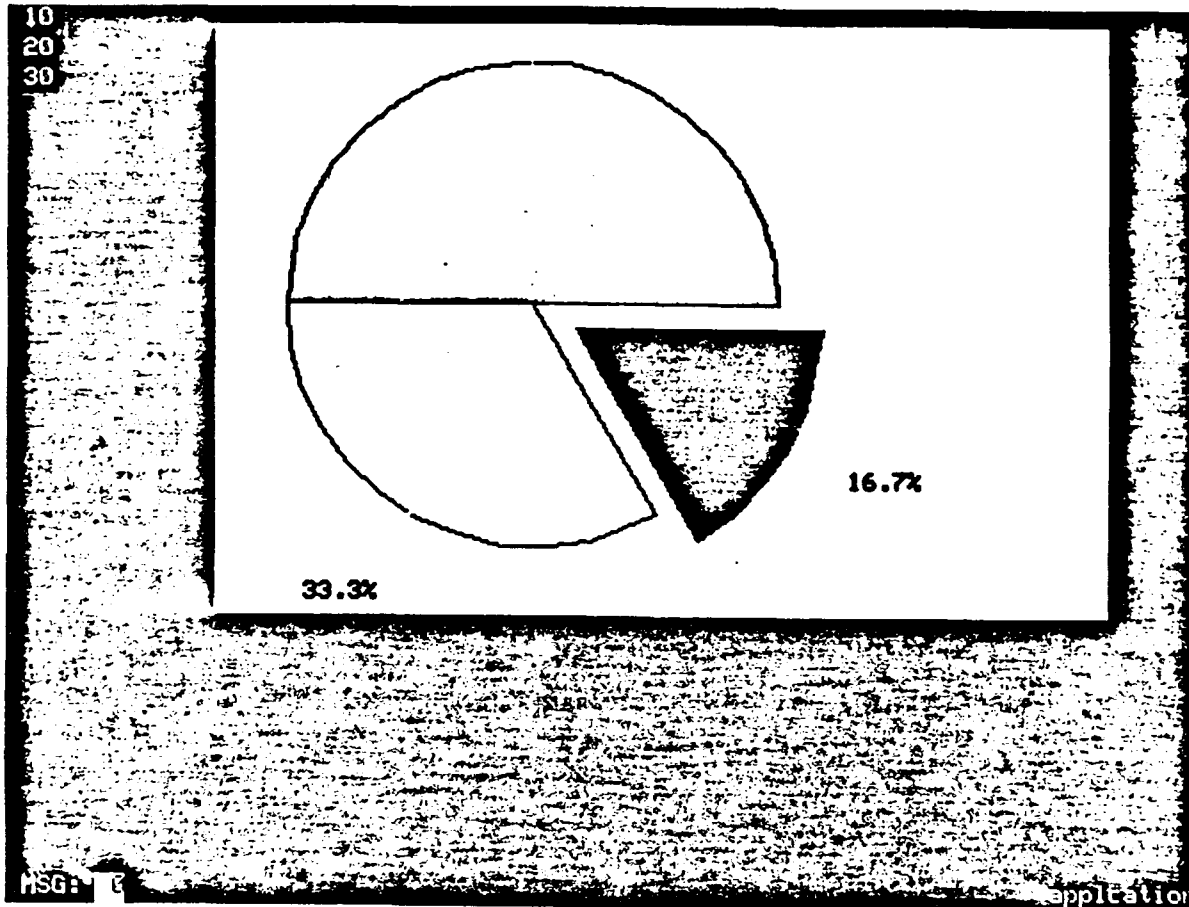
create pie graph grf6
    at 10 2
    size 40 by 16
    using ('grftst7.i1')

pie 1
    shade color red

pie 2
    shade color magenta

pie 3
    shade color white
    explode 20
```

Figure A-8 and corresponding GDL



```
create form grftst8
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

graph grf7
    at 1 15
    size 60 by 20
    display as blue

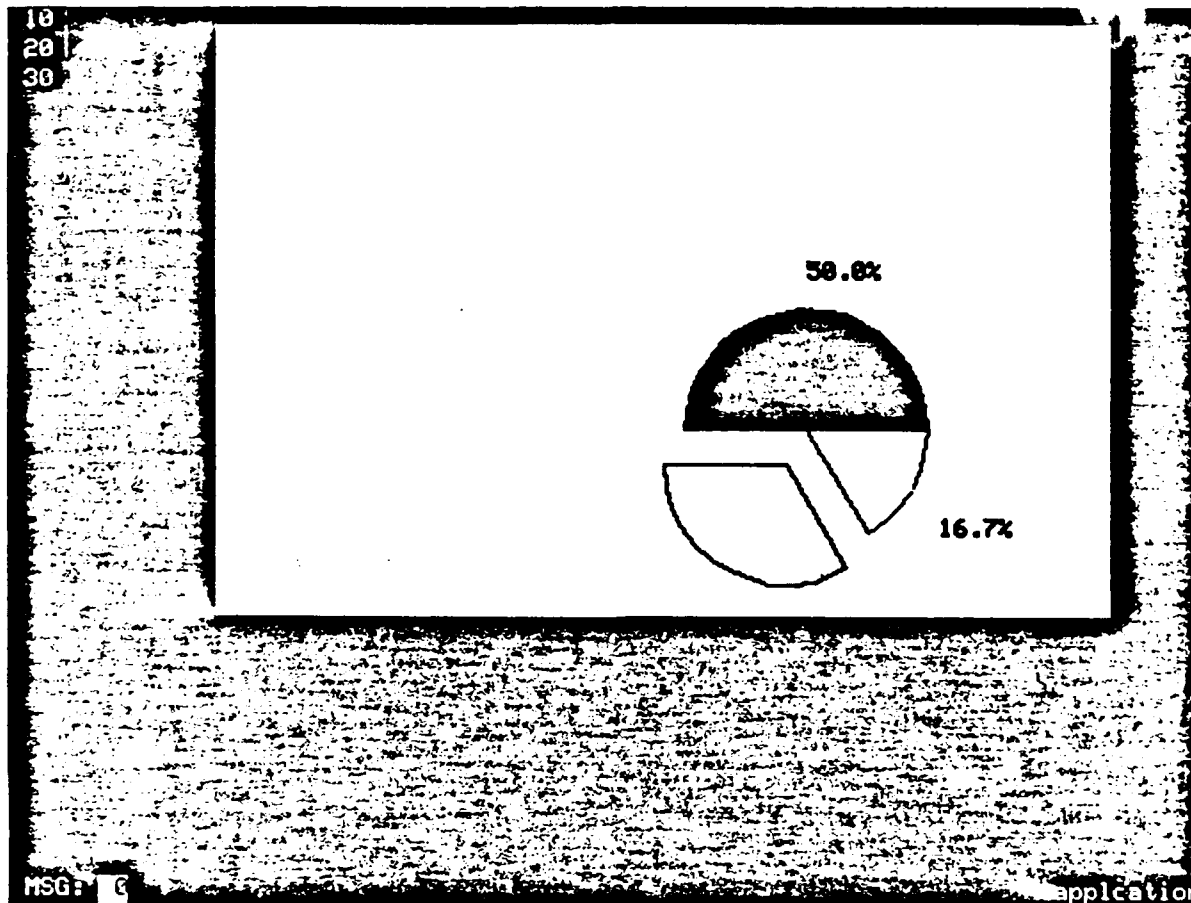
create pie graph grf7
    at 2 2
    size 40 by 16
    using ('grftst8.i1')

pie 1
    shade color yellow

pie 2
    shade color white

pie 3
    shade color red
    explode 20
```

Figure A-9 and corresponding GDL



```
create form grftst9
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

graph grf8
    at 1 15
    size 60 by 20
    display as blue

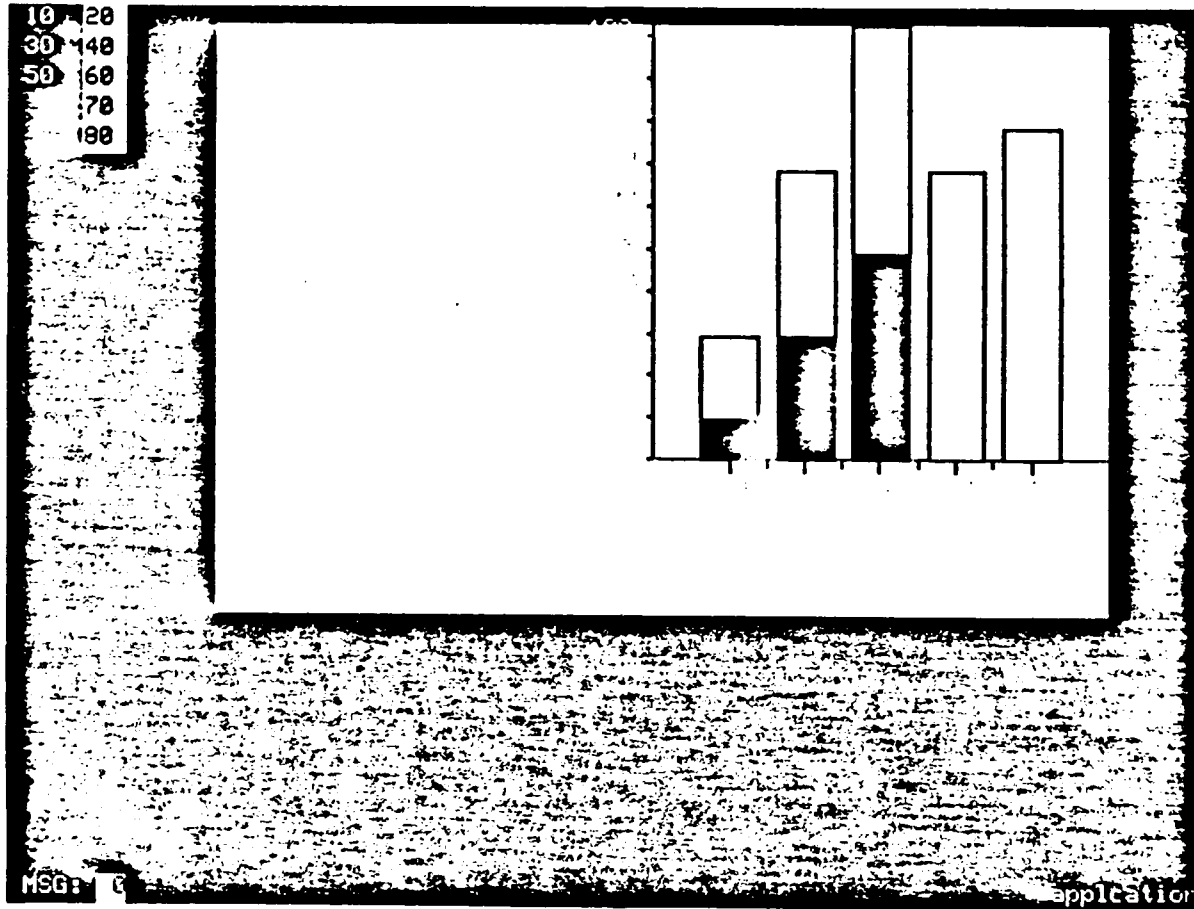
create pie graph grf8
    at 10 30
    size 20 by 8
    using ('grftst9.i1')

pie 1
    shade color red

pie 2
    shade color magenta
    explode 35

pie 3
    shade color white
```

Figure A-10 and corresponding GDL



```
create form grftst10
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf9
    at 1 15
    size 60 by 20
    display as blue

create bar graph grf9
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    background blue

curve aaa
    'grftst10.i1' using axis ax2
    absolute

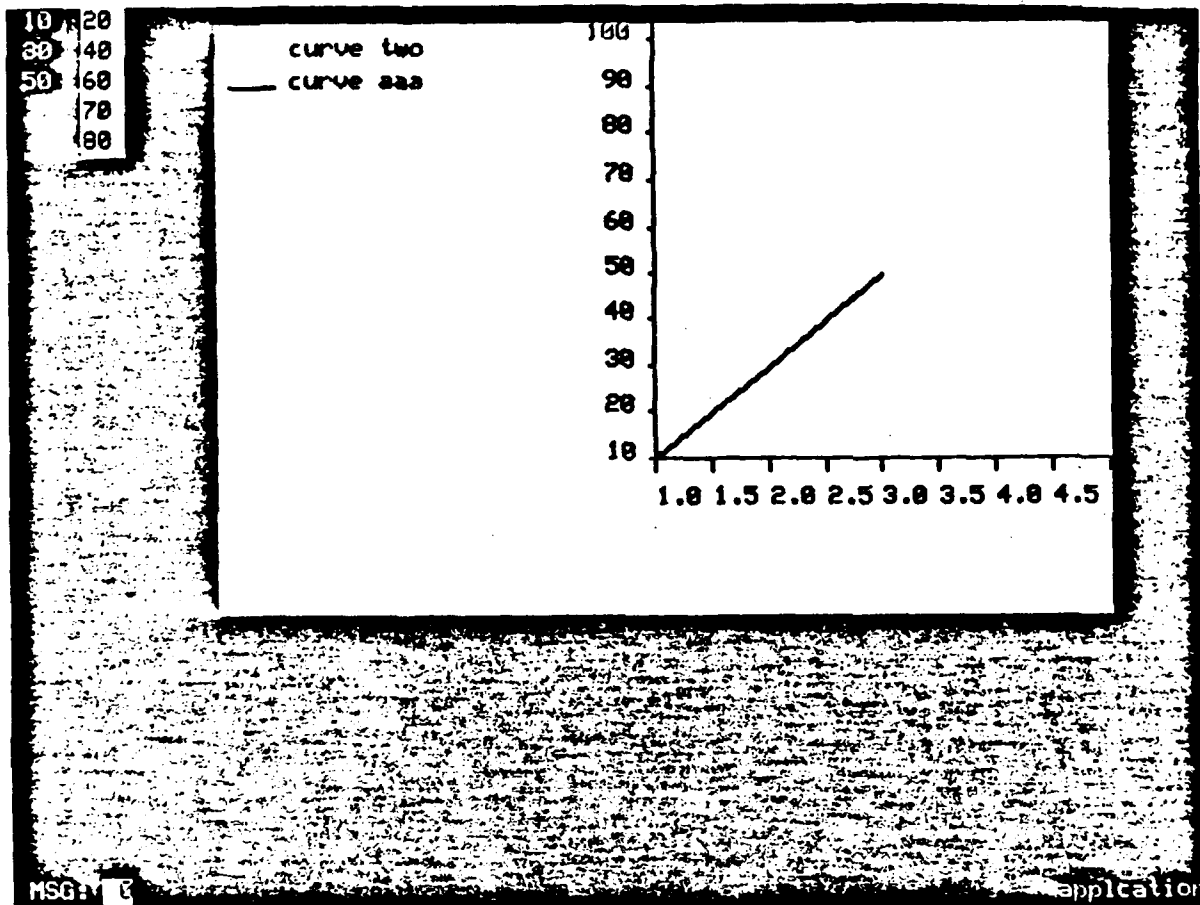
curve two
    'grftst10.i2'
    additive using curve aaa

axis ax1
    horizontal
    display as x
    at 15 30
    tick 5 1 c "A" "B" "C"
    size 30
```


UTP620344210
30 September 1990

axis ax2
at 15 30
size 15
vertical
display as x

Figure A-11 and corresponding GDL



```
create form grftst11
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

item i2 (5 v 0)
  display as yellow
  at 1 6
  size 3
  domain (numeric)

graph grf10
  at 1 15
  size 60 by 20
  display as blue

create line graph grf10
  using (1, 2, 3, 4, 5 axis ax1)
  attribute a fill (display cyan)
  attribute b line (display magenta)
  attribute xy prompt (display yellow)
  attribute x line (display yellow)
  attribute c prompt (display white)
  attribute d line (display green)
  legend at 2 2
  background blue

curve aaa
  'grftst11.i1' using axis ax2
  absolute
  legend xy "curve aaa"

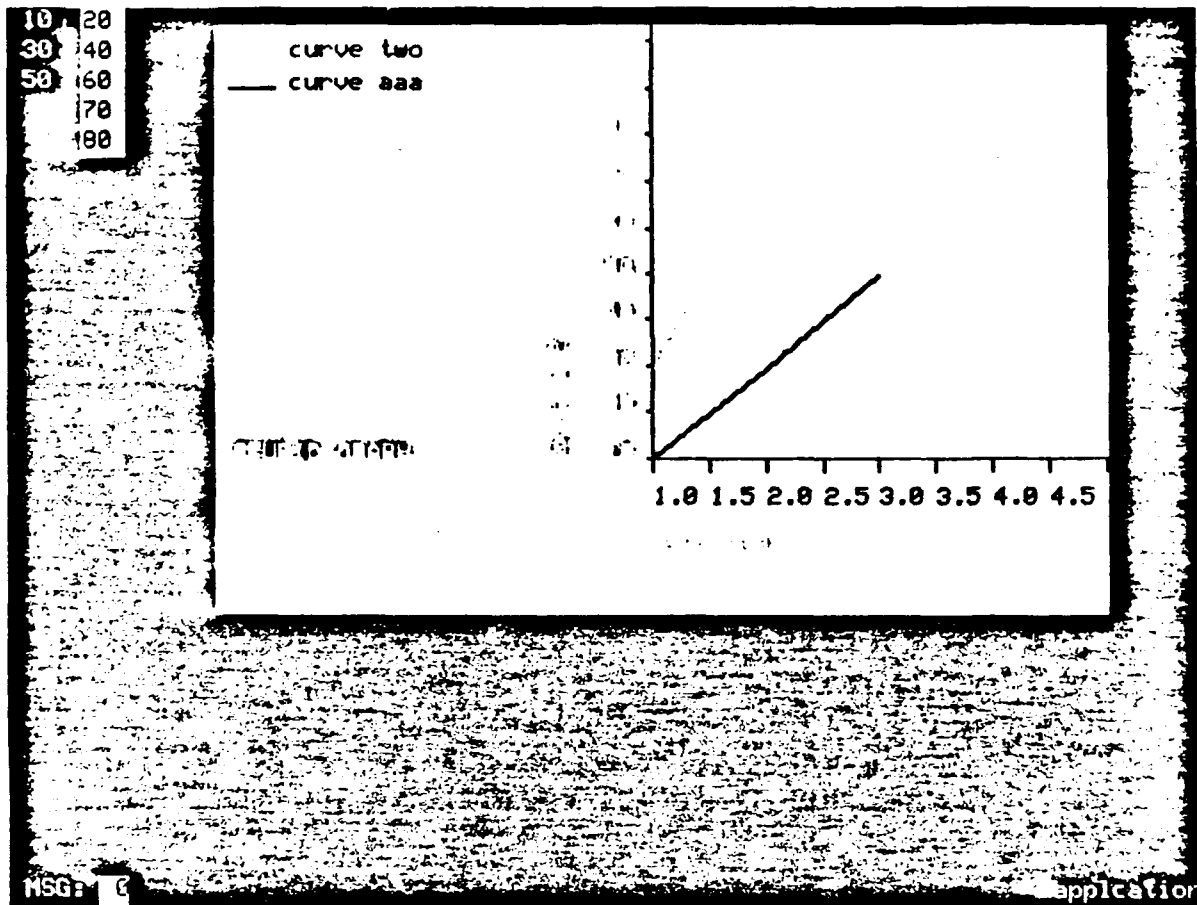
curve two
  'grftst11.i2'
  additive using curve aaa
  legend xy "curve two"

axis ax1
  horizontal
  display as x
  at 15 30
  size 30
```

UTP620344210
30 September 1990

axis ax2
at 15 30
size 15
vertical
display as x

Figure A-12 and corresponding GDL



```
create form grftst12
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf11
    at 1 15
    size 60 by 20
    display as blue

create line graph grf11
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    legend at 2 2
    label display as c, at 15 2 "STUPID GRAPH"
    background blue

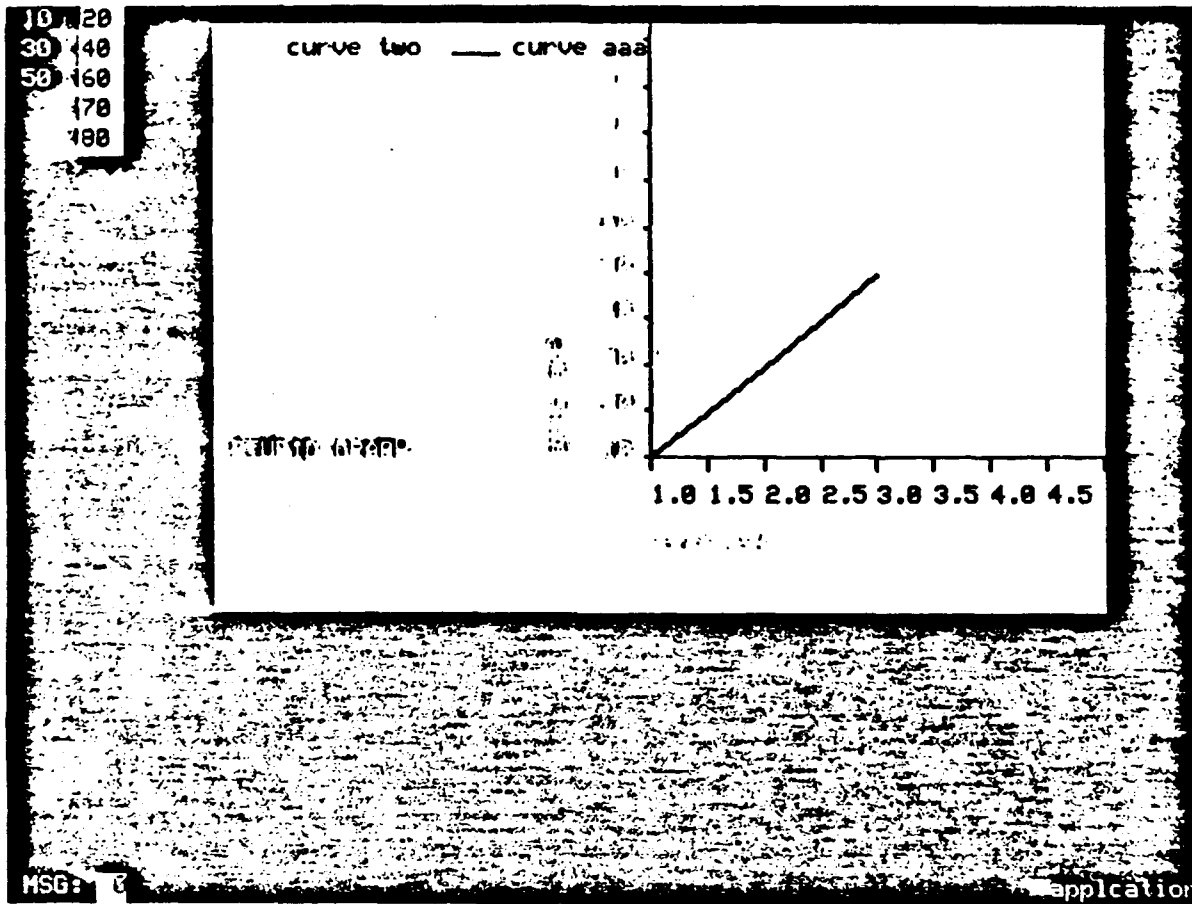
curve aaa
    'grftst12.i1' using axis ax2
    absolute
    legend xy "curve aaa"

curve two
    'grftst12.i2'
    additive using curve aaa
    legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-13 and corresponding GDL




```
create form grftstl3
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf12
    at 1 15
    size 60 by 20
    display as blue

create line graph grf12
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    legend h at 2 2
    label display as c, at 15 2 "STUPID GRAPH"
    background blue

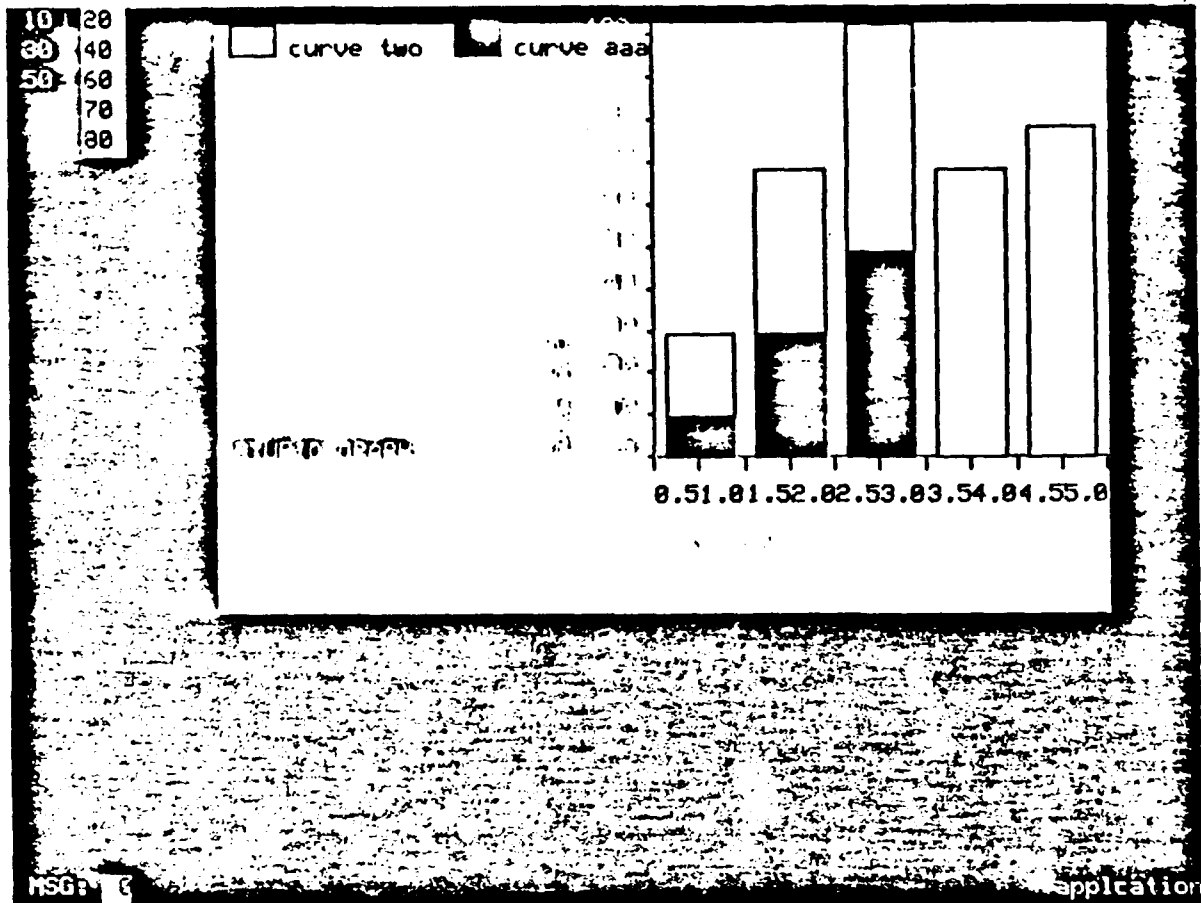
curve aaa
    'grftstl3.i1' using axis ax2
    absolute
    legend xy "curve aaa"

curve two
    'grftstl3.i2'
    additive using curve aaa
    legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-14 and corresponding GDL



```
create form grftst14
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf13
    at 1 15
    size 60 by 20
    display as blue

create bar graph grf13
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    legend h at 2 2
    label display as c, at 15 2 "STUPID GRAPH"
    background blue

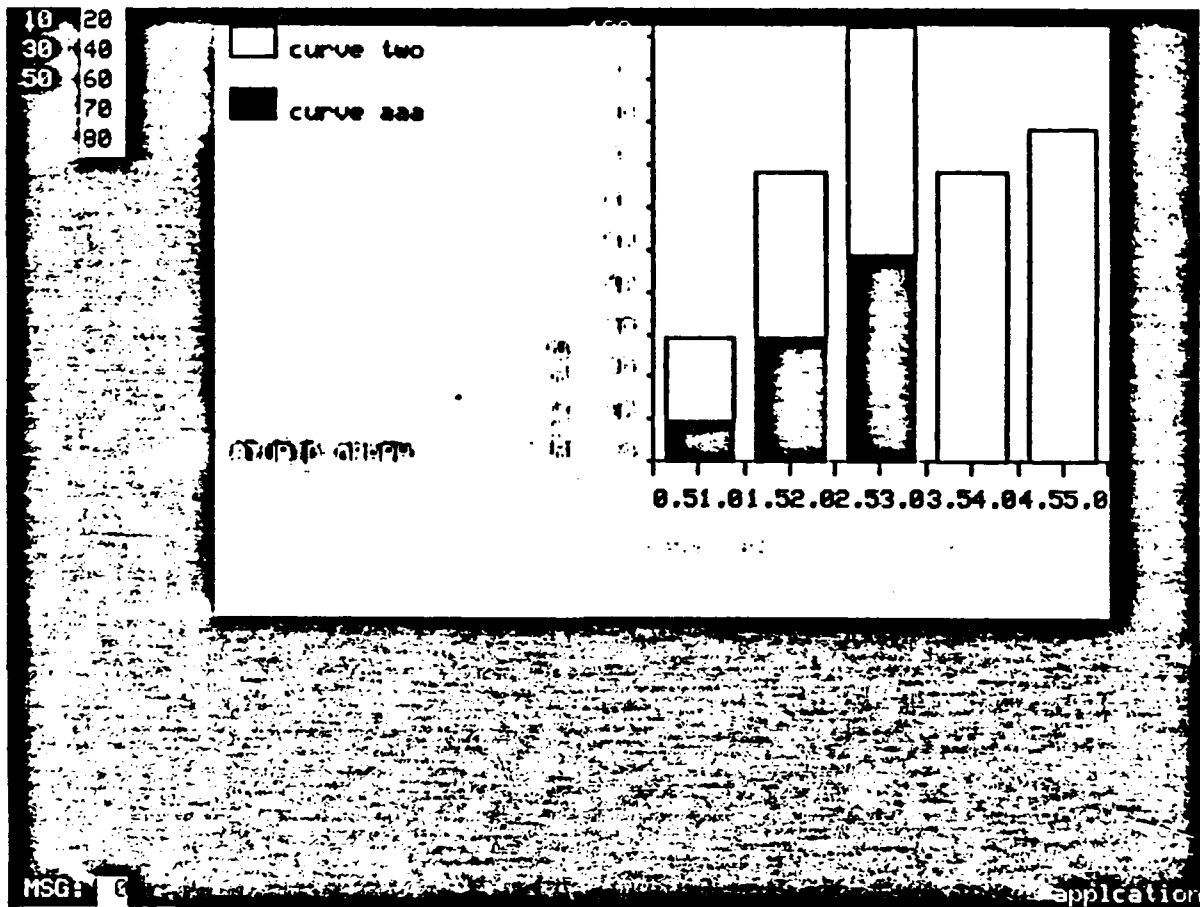
curve aaa
    'grftst14.i1' using axis ax2
    absolute
    legend xy "curve aaa"

curve two
    'grftst14.i2'
    additive using curve aaa
    legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-15 and corresponding GDL



```
create form grftst15
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf14
    at 1 15
    size 60 by 20
    display as blue

create bar graph grf14
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    legend at 2 2
    label display as c, at 15 2 "STUPID GRAPH"
    background blue

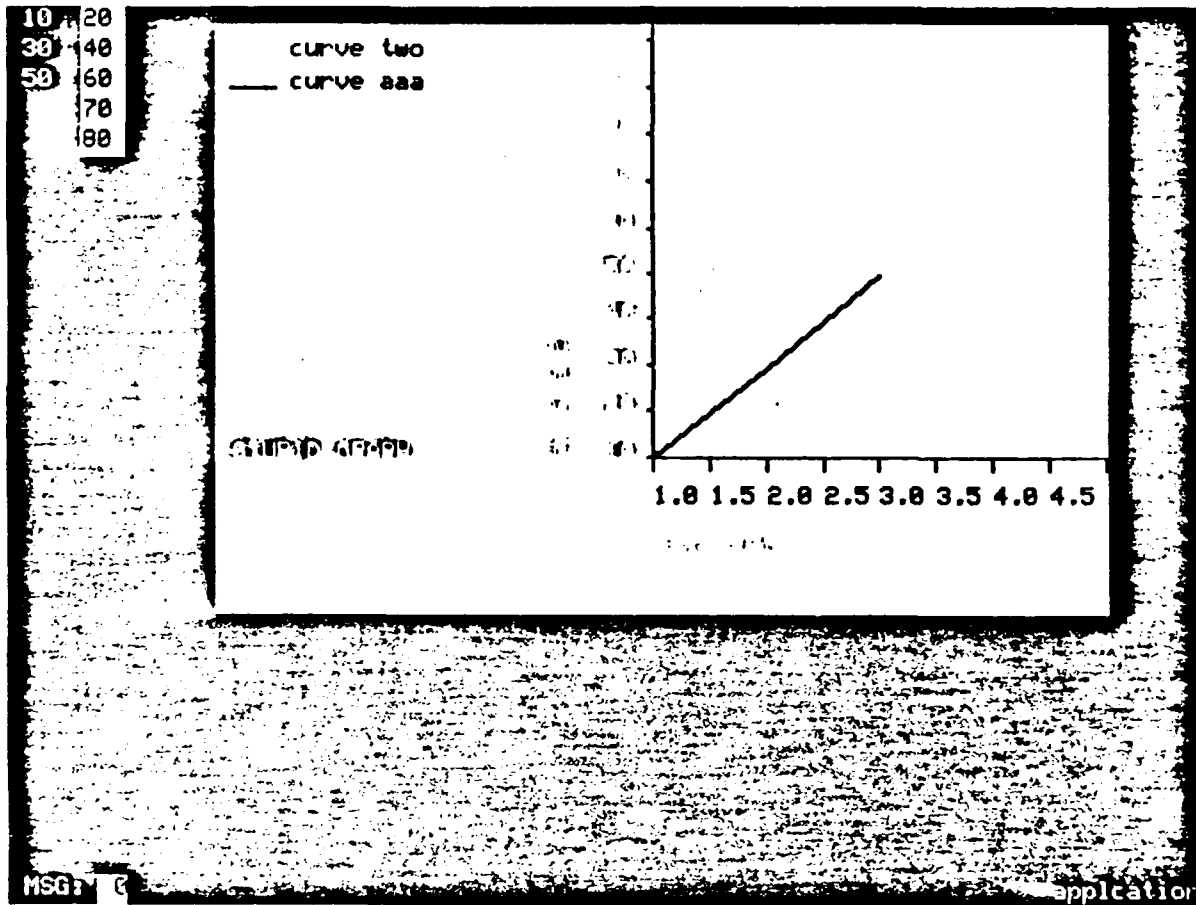
curve aaa
    'grftst15.i1' using axis ax2
    absolute
    legend xy "curve aaa"

curve two
    'grftst15.i2'
    additive using curve aaa
    legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```


Figure A-16 and corresponding GDL



```
create form grftst16
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

item i2 (5 v 0)
  display as yellow
  at 1 6
  size 3
  domain (numeric)

graph grf15
  at 1 15
  size 60 by 20
  display as blue

create line graph grf15
  using (1, 2, 3, 4, 5 axis ax1)
  attribute a fill (display cyan)
  attribute b line (display magenta)
  attribute xy prompt (display yellow)
  attribute x line (display yellow)
  attribute c prompt (display white)
  attribute d line (display green)
  legend at 2 2 box
  label display as c, at 15 2 "STUPID GRAPH"
  background blue

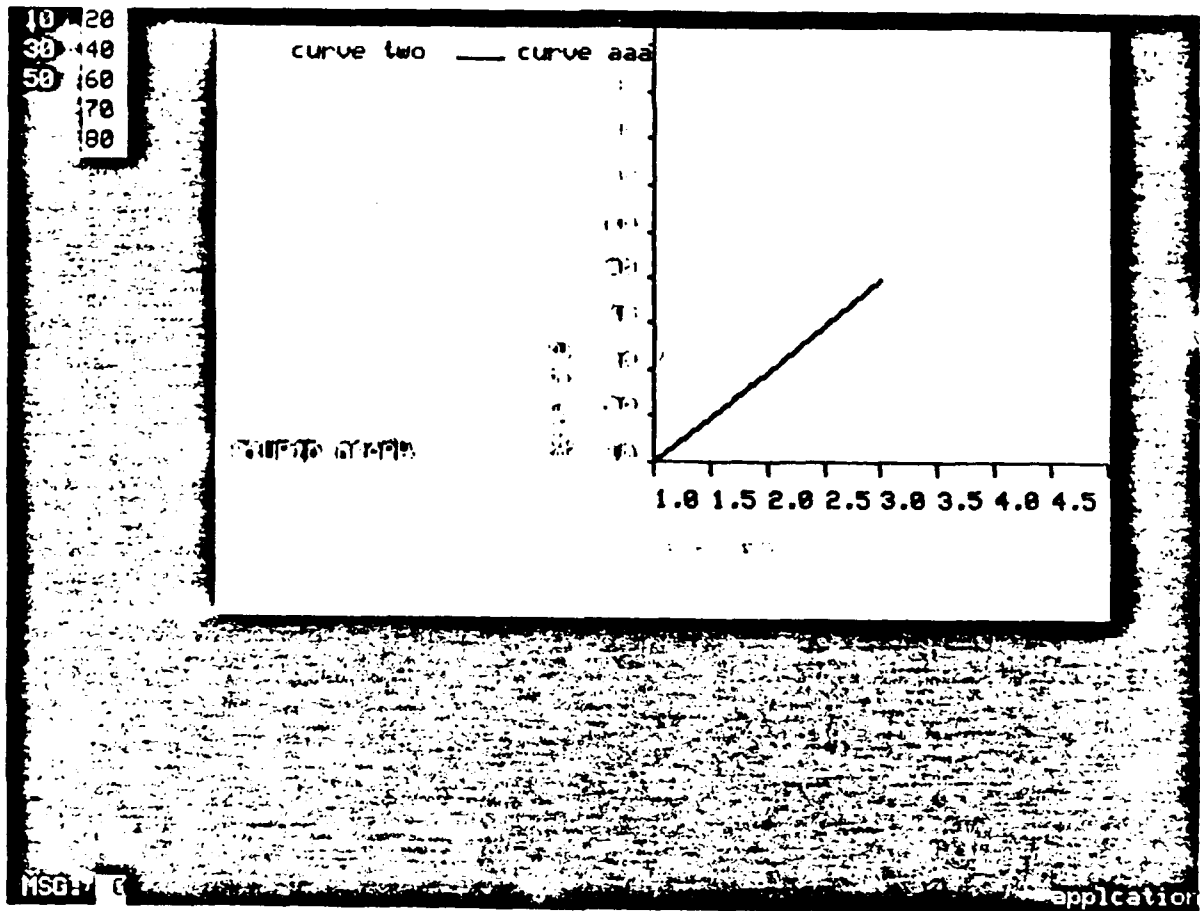
curve aaa
  'grftst16.i1' using axis ax2
  absolute
  legend xy "curve aaa"

curve two
  'grftst16.i2'
  additive using curve aaa
  legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-17 and corresponding GDL



```
create form grftst17
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

item i2 (5 v 0)
  display as yellow
  at 1 6
  size 3
  domain (numeric)

graph grf16
  at 1 15
  size 60 by 20
  display as blue

create line graph grf16
  using (1, 2, 3, 4, 5 axis ax1)
  attribute a fill (display cyan)
  attribute b line (display magenta)
  attribute xy prompt (display yellow)
  attribute x line (display yellow)
  attribute c prompt (display white)
  attribute d line (display green)
  legend h at 2 2 box
  label display as c, at 15 2 "STUPID GRAPH"
  background blue

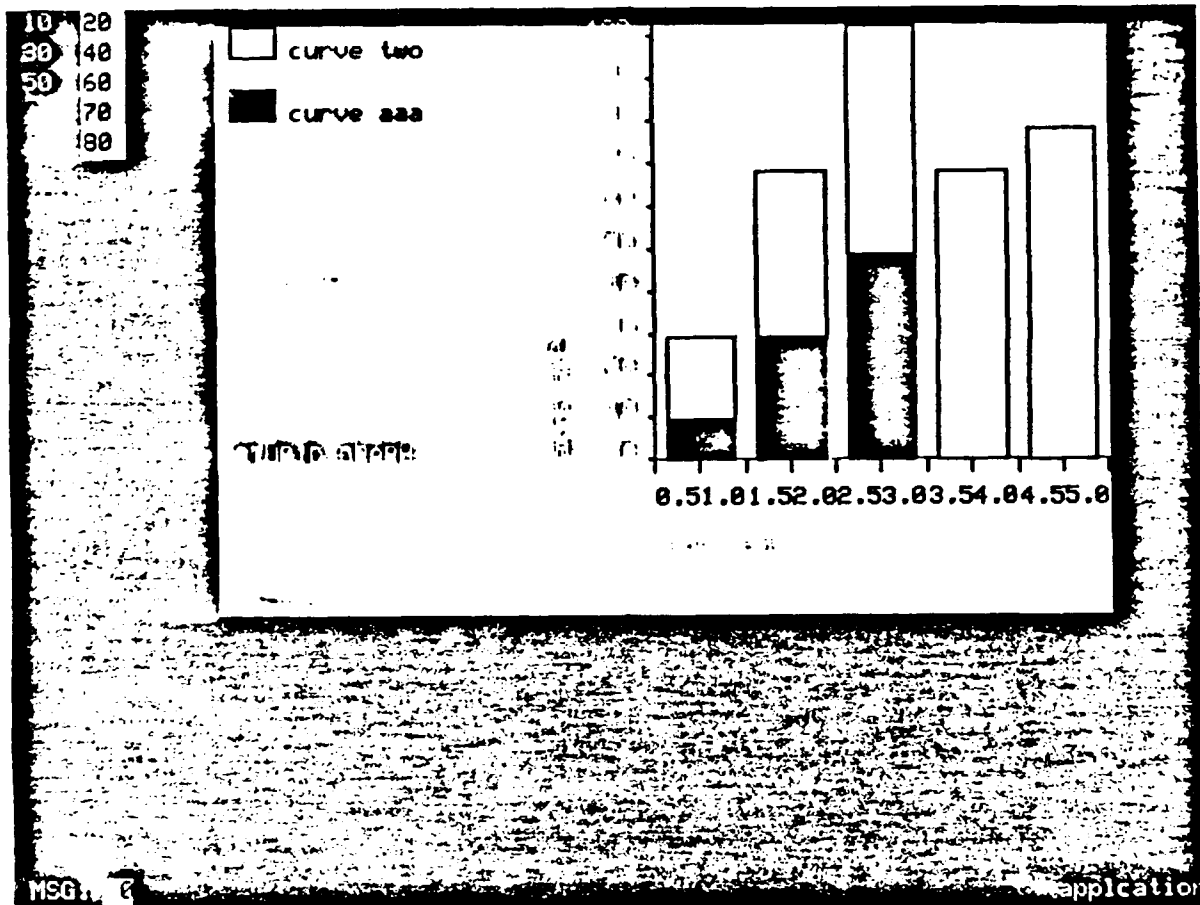
curve aaa
  'grftst17.i1' using axis ax2
  absolute
  legend xy "curve aaa"

curve two
  'grftst17.i2'
  additive using curve aaa
  legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-18 and corresponding GDL



```
create form grftst18
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

item i2 (5 v 0)
  display as yellow
  at 1 6
  size 3
  domain (numeric)

graph grf17
  at 1 15
  size 60 by 20
  display as blue

create bar graph grf17
  using (1, 2, 3, 4, 5 axis ax1)
  attribute a fill (display cyan)
  attribute b line (display magenta)
  attribute xy prompt (display yellow)
  attribute x line (display yellow)
  attribute c prompt (display white)
  attribute d line (display green)
  legend at 2 2 box
  label display as c, at 15 2 "STUPID GRAPH"
  background blue

curve aaa
  'grftst18.i1' using axis ax2
  absolute
  legend xy "curve aaa"

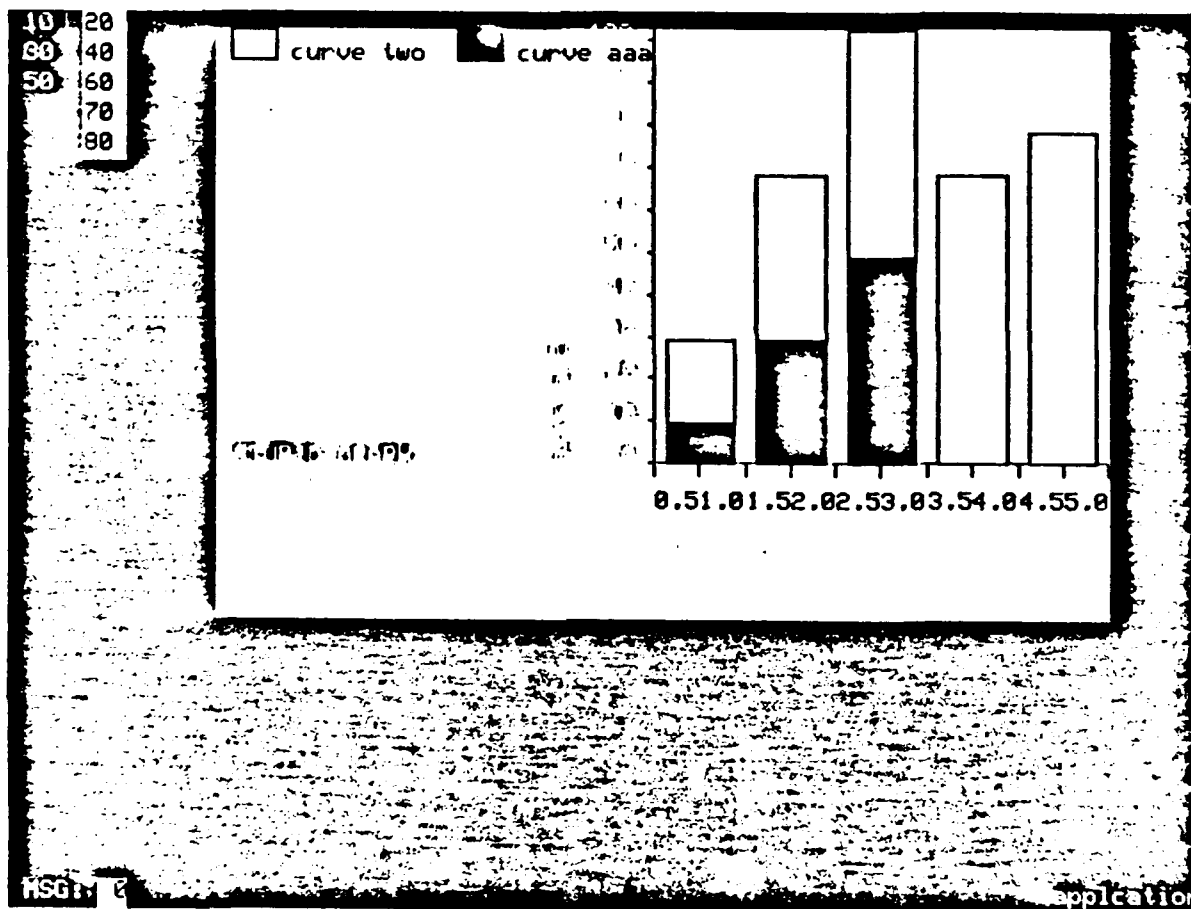
curve two
  'grftst18.i2'
  additive using curve aaa
  legend xy "curve two"
```



```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-19 and corresponding GDL



```
create form grftst19
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf18
    at 1 15
    size 60 by 20
    display as blue

create bar graph grf18
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    legend h at 2 2 box
    label display as c, at 15 2 "STUPID GRAPH"
    background blue

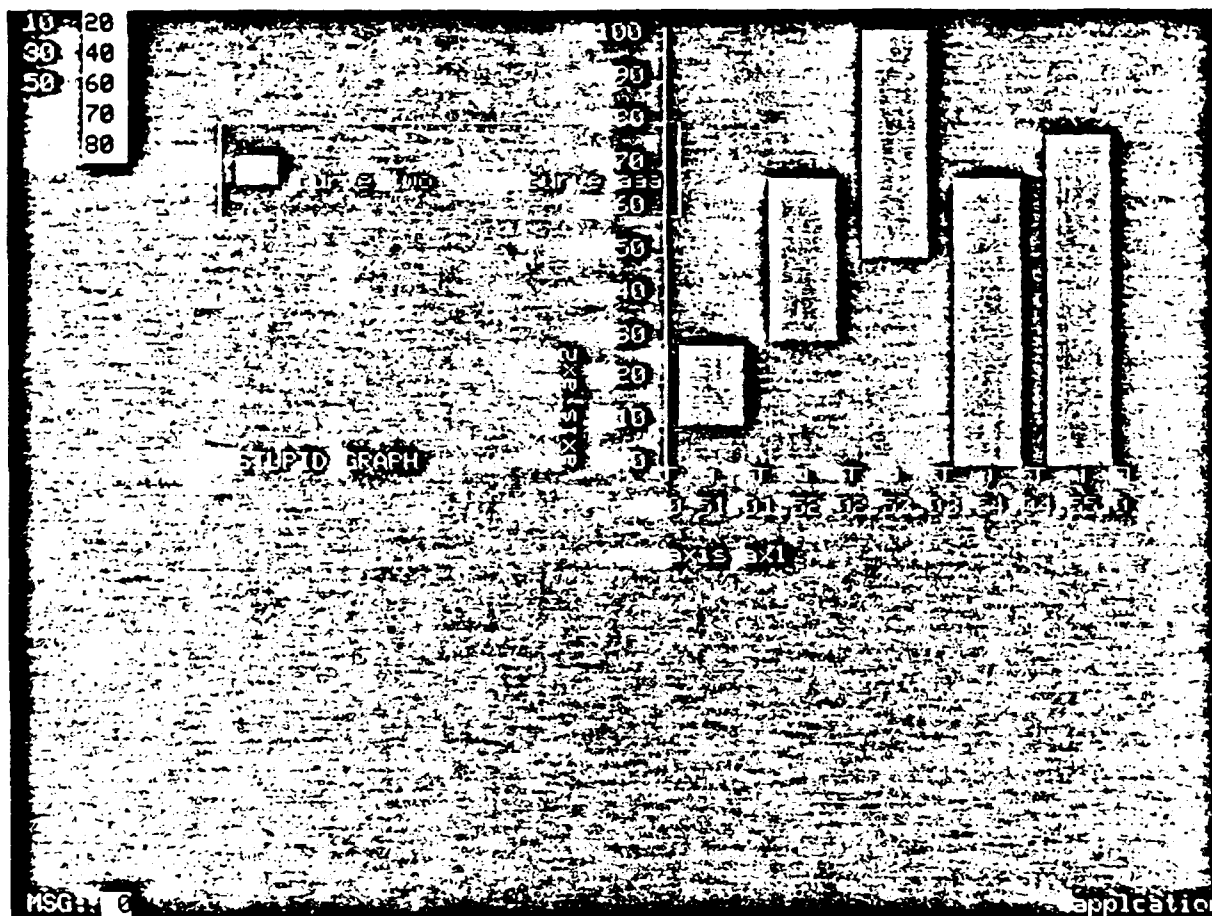
curve aaa
    'grftst19.i1' using axis ax2
    absolute
    legend xy "curve aaa"

curve two
    'grftst19.i2'
    additive using curve aaa
    legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-20 and corresponding GDL



```
create form grftst20
    size 80 by 30

tem i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf19
    at 1 15
    size 60 by 20
    display as blue

create bar graph grf19
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a fill (display cyan)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    legend h at 6 2 box
    label display as c, at 15 2 "STUPID GRAPH"
    background blue

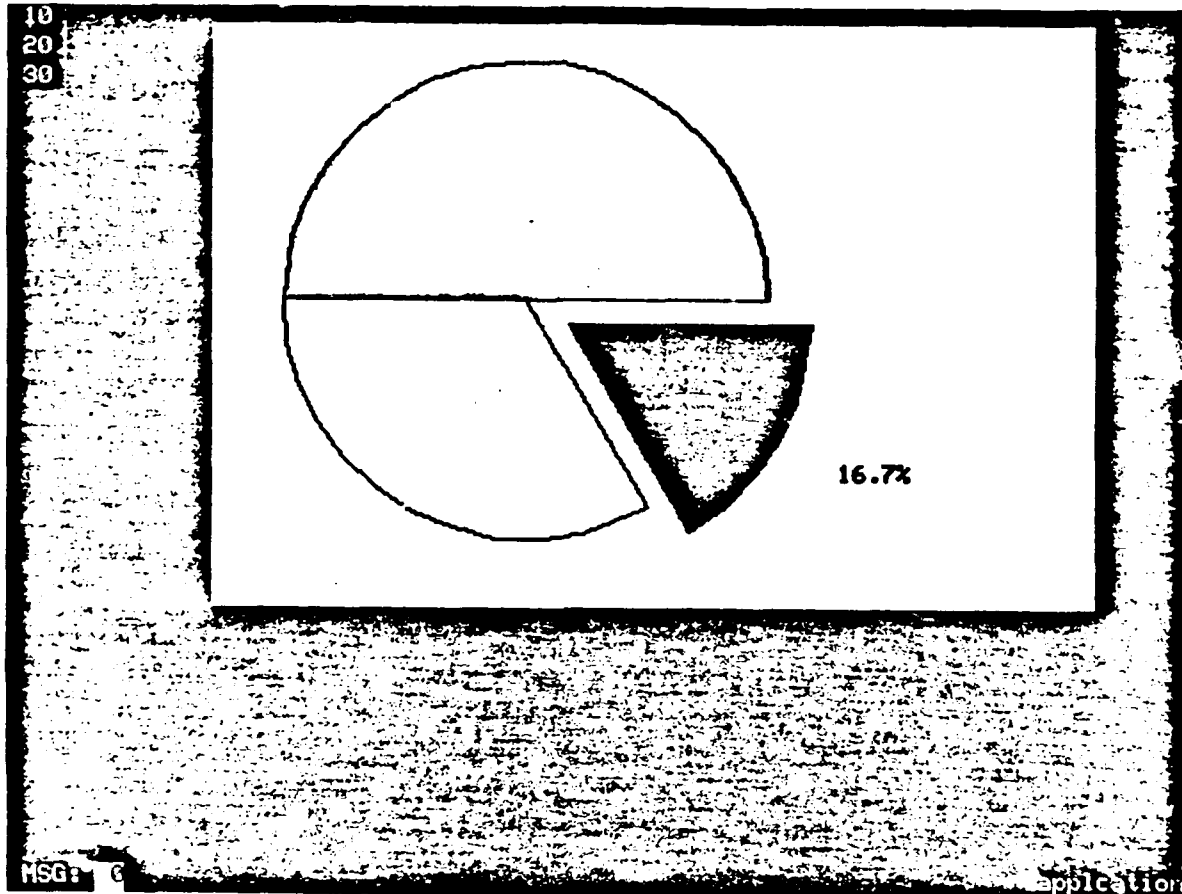
curve aaa
    'grftst20.i1' using axis ax2
    absolute
    legend xy "curve aaa"

curve two
    'grftst20.i2'
    additive using curve aaa
    legend xy "curve two"
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  size 30
  label c "axis ax1"
```

```
axis ax2
  at 15 30
  size 15
  label c "axis ax2"
  vertical
  display as x
```

Figure A-21 and corresponding GDL




```
create form grftst21  
  size 80 by 30
```

```
item i1 (3 v 0)  
  display as red  
  at 1 2  
  size 3  
  domain (numeric)
```

```
graph grf20  
  at 1 15  
  size 60 by 20  
  display as blue
```

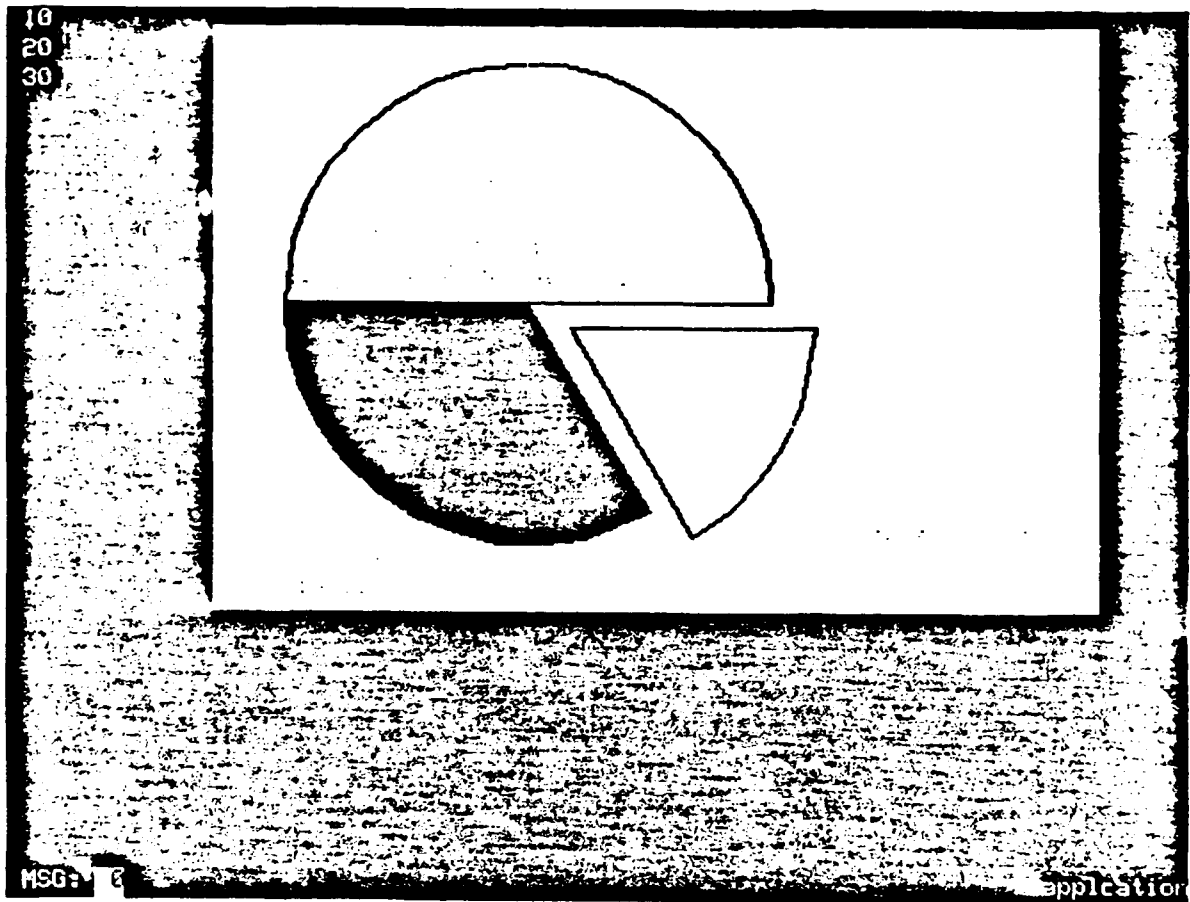
```
create pie graph grf20  
  at 2 2  
  size 40 by 16  
  using ('grftst21.i1')  
  attribute c prompt (display magenta)
```

```
pie 1  
  quantity c outside  
  shade color yellow
```

```
pie 2  
  percent c inside  
  quantity c outside  
  shade color white
```

```
pie 3  
  shade color red  
  explode 20
```

Figure A-22 and corresponding GDL



```
create form grftst22
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

graph grf21
  at 1 15
  size 60 by 20
  display as blue

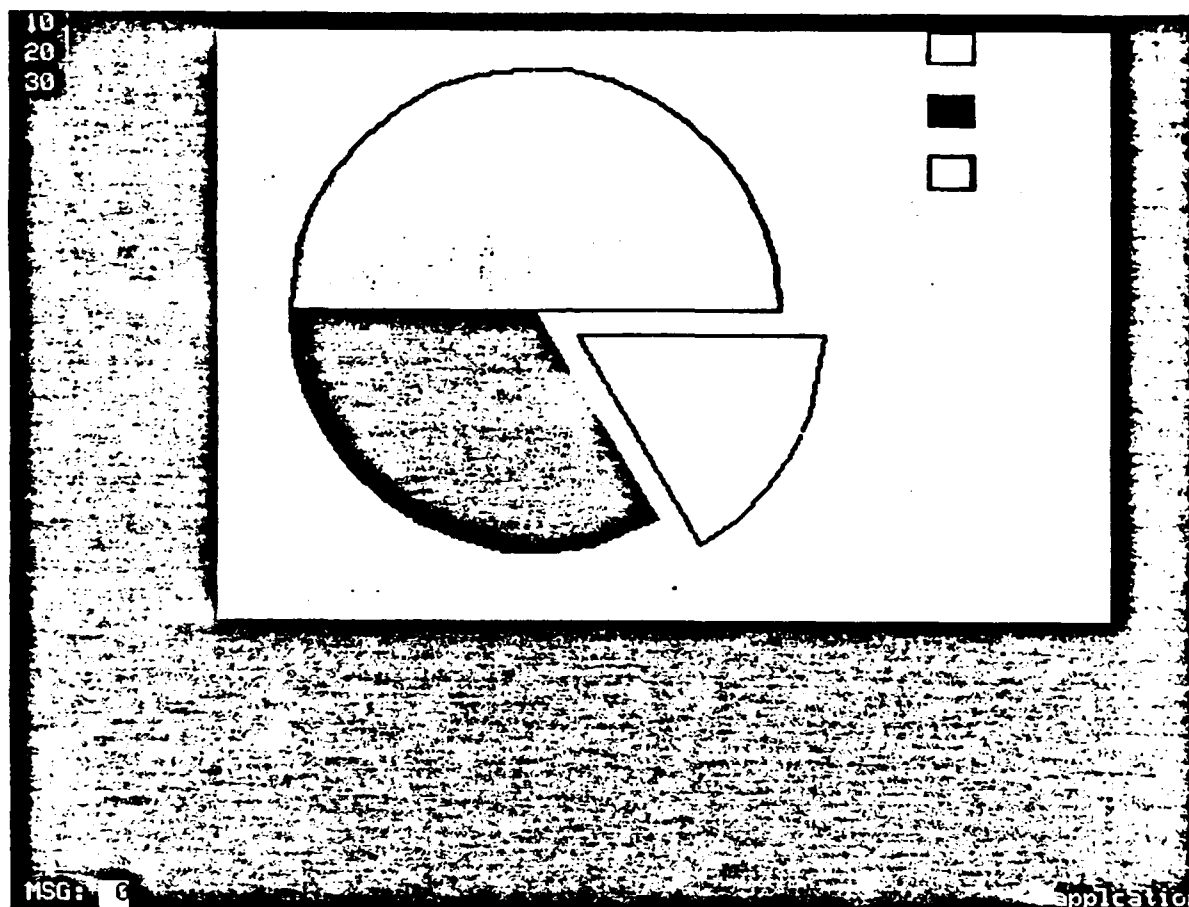
create pie graph grf21
  at 2 2
  size 40 by 16
  using ('grftst22.i1')
  attribute c prompt (display magenta)

pie 1
  quantity c outside
  shade color yellow

pie 2
  shade color red

pie 3
  percent c outside
  quantity c outside
  label c "this is a white pie slice"
  shade color white
  explode 20
```

Figure A-23 and corresponding GDL



```
create form grftst23
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

graph grf22
    at 1 15
    size 60 by 20
    display as blue

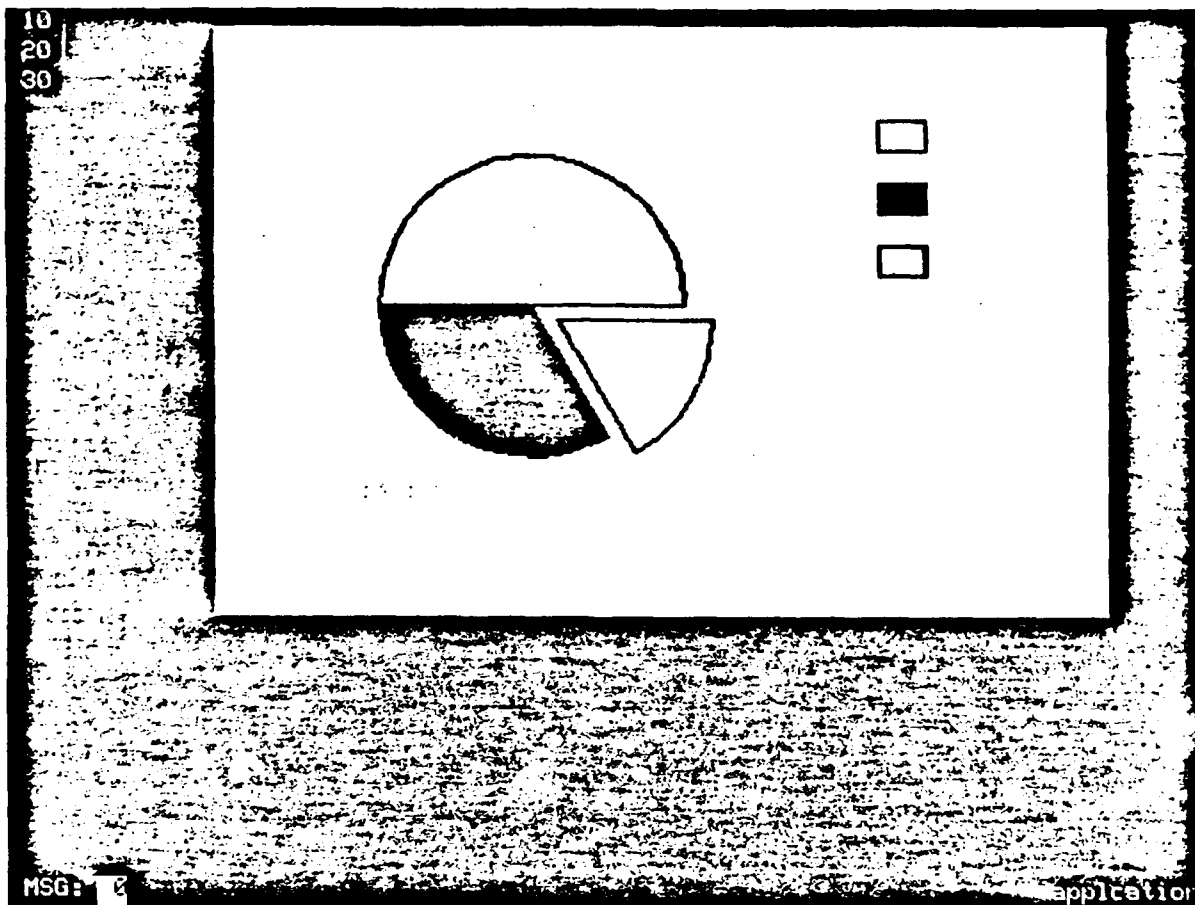
create pie graph grf22
    at 2 2
    size 40 by 16
    using ('grftst23.i1')
    attribute c prompt (display magenta)
    legend at 2 48 box

pie 1
    quantity c outside
    shade color yellow
    legend c "segment 1"

pie 2
    shade color red
    legend c "segment 2"

pie 3
    percent c outside
    quantity c outside
    shade color white
    legend c "segment 3"
    explode 20
```

Figure A-24 and corresponding GDL



```
create form grftst24
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

graph grf23
  at 1 15
  size 60 by 20
  display as blue

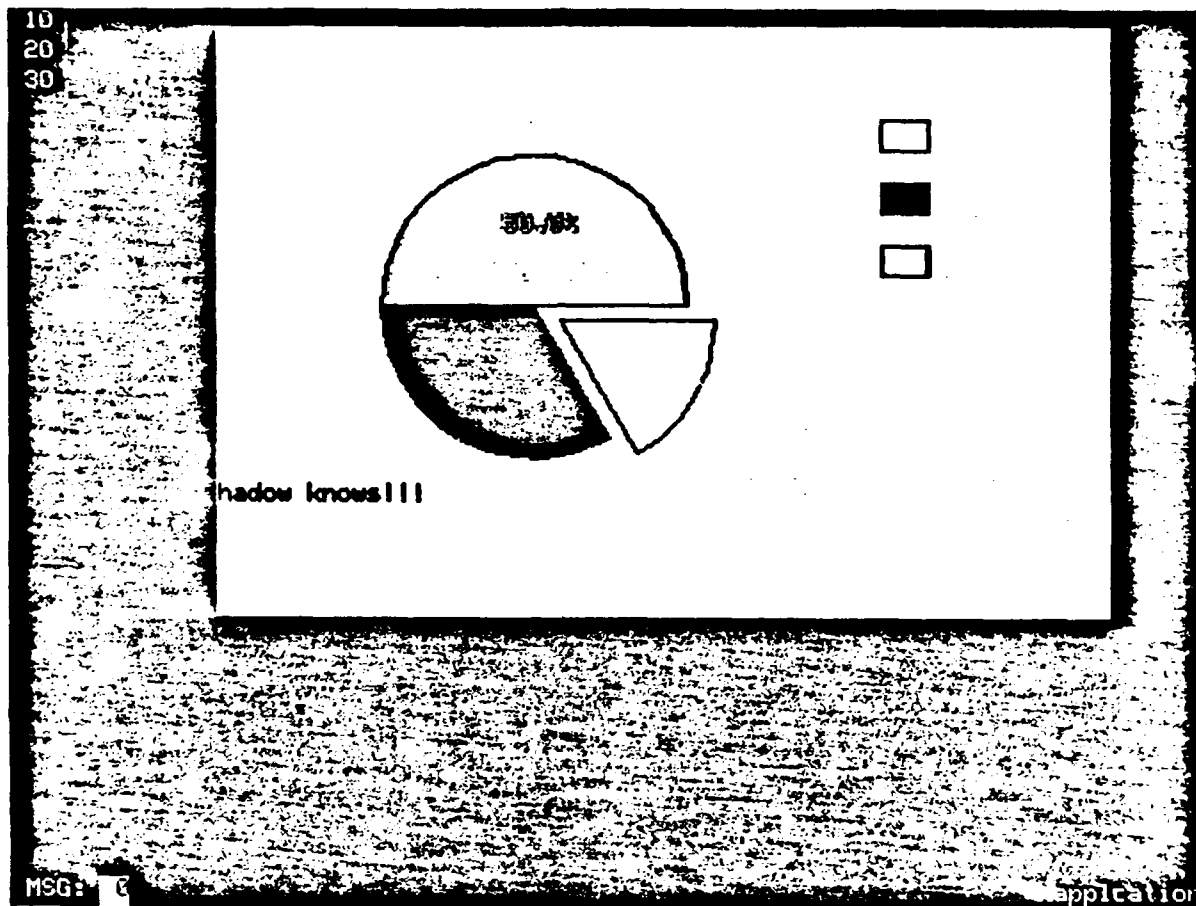
create pie graph grf23
  at 5 2
  size 40 by 10
  using ('grftst24.i1')
  attribute c prompt (display magenta)
  legend at 5 45 box

pie 1
  quantity c outside
  shade color yellow
  legend c "segment 1"

pie 2
  shade color red
  legend c "segment 2"

pie 3
  percent c outside
  quantity c outside
  shade color white
  legend c "segment 3"
  explode 20
```

Figure A-25 and corresponding GDL




```
create form grftst25
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

graph grf24
  at 1 15
  size 60 by 20
  display as blue

create pie graph grf24
  at 5 2
  size 40 by 10
  using ('grftst25.i1')
  attribute a prompt (display red)
  attribute b prompt (display blue)
  attribute c prompt (display magenta)
  legend at 5 45 box

pie 1
  quantity c outside
  shade color yellow
  percent b inside
  legend c "segment 1"

pie 2
  shade color red
  label a "The Shadow knows!!!"
  legend c "segment 2"

pie 3
  percent c outside
  quantity c outside
  shade color white
  label c "Who knows what evil lurks"
  label c "  in the hearts of men?"
  legend c "segment 3"
  explode 20
```

Figure A-26 and corresponding GDL



```
create form grftst26
  size 80 by 30

item i1 (3 v 0)
  display as red
  at 1 2
  size 3
  domain (numeric)

item i2 (5 v 0)
  display as yellow
  at 1 6
  size 3
  domain (numeric)

graph grf25
  at 1 15
  size 60 by 20
  display as blue

create bar graph grf25
  using (1, 2, 3, 4, 5 axis ax1)
  attribute a fill (display cyan)
  attribute b line (display magenta)
  attribute xy prompt (display yellow)
  attribute x line (display yellow)
  attribute c prompt (display white)
  attribute d line (display green)
  background blue

curve aaa
  'grftst26.i1' using axis ax2
  absolute

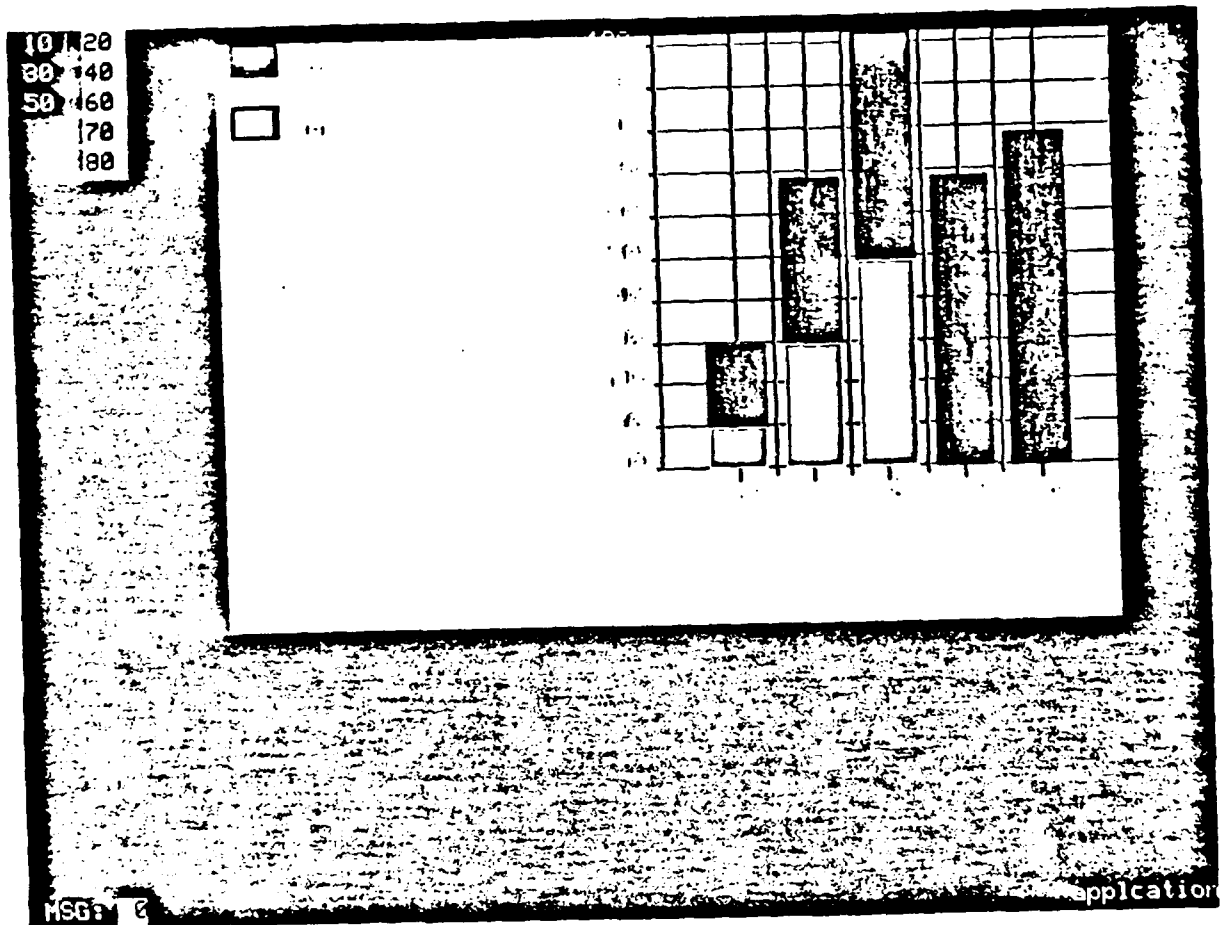
curve two
  'grftst26.i2'
  additive using curve aaa

axis ax1
  horizontal
  display as x
  at 15 30
  size 30
```

UTP620344210
30 September 1990

axis ax2
at 15 30
size 15
vertical
display as x

Figure A-27 and corresponding GDL



```
create form grftst27
    size 80 by 30

item i1 (3 v 0)
    display as red
    at 1 2
    size 3
    domain (numeric)

item i2 (5 v 0)
    display as yellow
    at 1 6
    size 3
    domain (numeric)

graph grf26
    at 1 15
    size 60 by 20
    display as blue

create bar graph grf26
    using (1, 2, 3, 4, 5 axis ax1)
    attribute a line (display cyan)
    attribute e line (display red)
    attribute b line (display magenta)
    attribute xy prompt (display yellow)
    attribute x line (display yellow)
    attribute c prompt (display white)
    attribute d line (display green)
    legend at 2 2
    background blue

curve aaa
    'grftst10.i1' using axis ax2
    absolute
    shade color yellow
    display as a

curve two
    'grftst10.i2'
    additive using curve aaa
    shade color green
    display as e
```

```
axis ax1
  horizontal
  display as x
  at 15 30
  tick 5 1 c "A" "B" "C"
  size 30
  fine grid
```

```
axis ax2
  at 15 30
  size 15
  vertical
  display as x
  grid
  maximum 110
```

APPENDIX B

SCREENS AND GDL FOR GRAFDE

This appendix contains all the screens for the second test of the Graph Definition Language. The corresponding GDL follows the screens. Also included is the ADL that defines the interactive portion of the application.

```
create application grafdemo

keypad ( entkey = 0 exitkey = 4 )

on ( startup() )
{
  present masterfrm
}

on ( pick ( pie1 ))
{
  present graffrm1 in 'grafwin'
}

on ( pick ( bar1 ))
{
  present graffrm2 in 'grafwin'
}

on ( pick ( line1 ))
{
  present graffrm3 in 'grafwin'
}

on ( pick ( pie2 ))
{
  present graffrm4 in 'grafwin'
}

on ( pick ( bar2 ))
{
  present graffrm5 in 'grafwin'
}
```



```
on ( pick ( line2 ))
{
    present graffrm6 in 'grafwin'
}

on ( pick ( exitkey ))
{
    exit
}

create form masterfrm

keypad (exitkey = 4 pie1 = 5 bar1 = 6 line1 = 7 pie2 = 9
bar2 = 10 line2 = 11)

    size 79 by 23
    prompt at 1 23
        "ENGINEERING CHANGE ANALYSIS CENTER"
    prompt at 2 32
        "CHANGE COST GRAPH"
    attribute BLAKGARD (background black, display yellow,
guarded, nowrite)

ITEM todatt
    at 1 68
    size 8
    prompt at 1 62 "DATE:"
    value '._date'
    display as BLAKGARD

ITEM curtime
    at 2 68
    size 8
    prompt at 2 62 "TIME:"
    value '._time'
    display as BLAKGARD

form userdat
    at 3 2
    size 78 by 6

window grafwin
    at 9 2
    size 78 by 15
    display as XPARNT
```

```
create form userdat
  size 78 by 6
  prompt at 1 11
    "Cur.Yr.      Cur.Yr.-1    Cur.Yr.-2    Cur.Yr.-3"
  prompt at 2 2
    "Rework:"
  prompt at 3 2
    "Scrap :"
  prompt at 4 2
    "Misc  :"
  prompt at 6 2
    "Enter data and press <PF5> - Pie ,  <PF6> - Bar,
<PF7> - Line"

form datafrm (4 h 5)
  at 2 10
  size 7 by 3

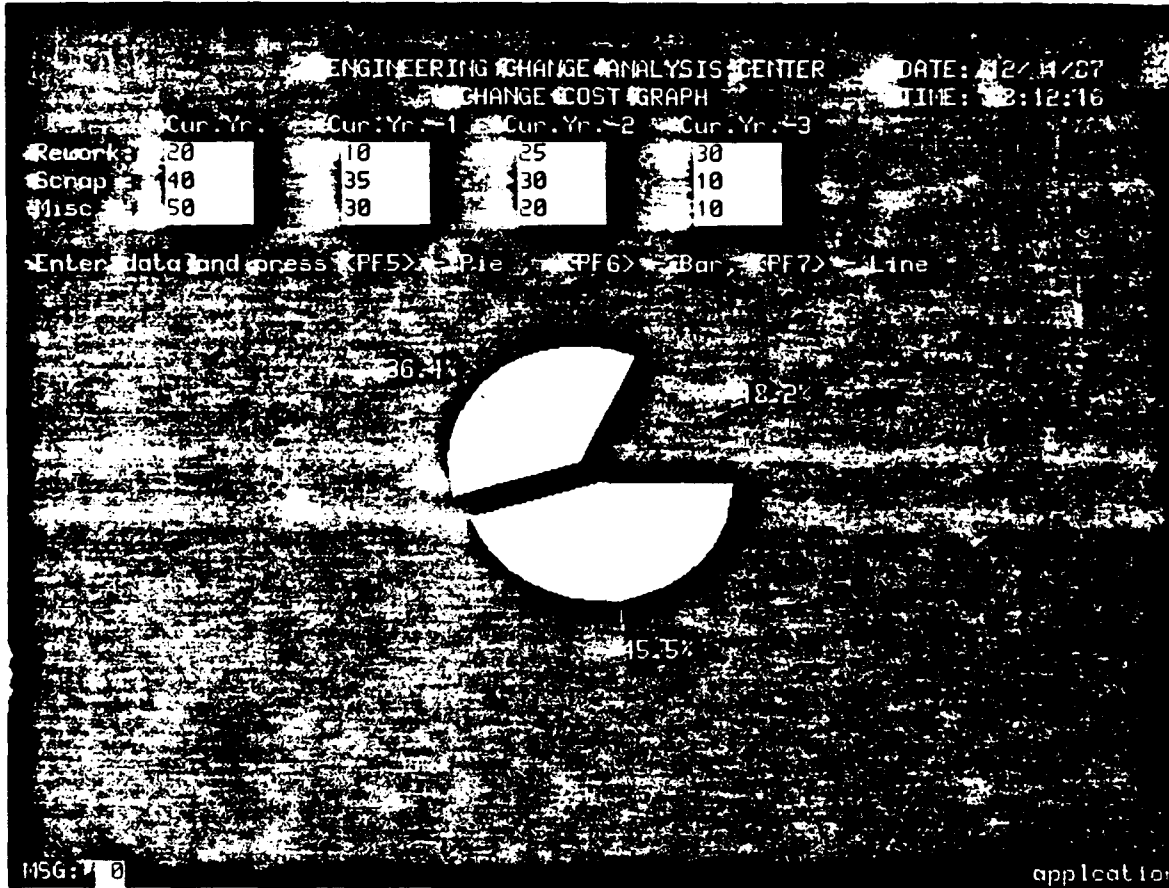
create form datafrm
  size 7 by 3

item rework
  at 1 2
  size 6
  display as INPUT
  domain ( numeric )

item scrap
  at 2 2
  size 6
  display as INPUT
  domain ( numeric )

item misc
  at 3 2
  size 6
  display as INPUT
  domain ( numeric )
```

Figure B-1 and corresponding GDL



```
create form graffrml
    size 79 by 15
graph piegraf
    at 1 1
    size 79 by 15
    display as XPARNT

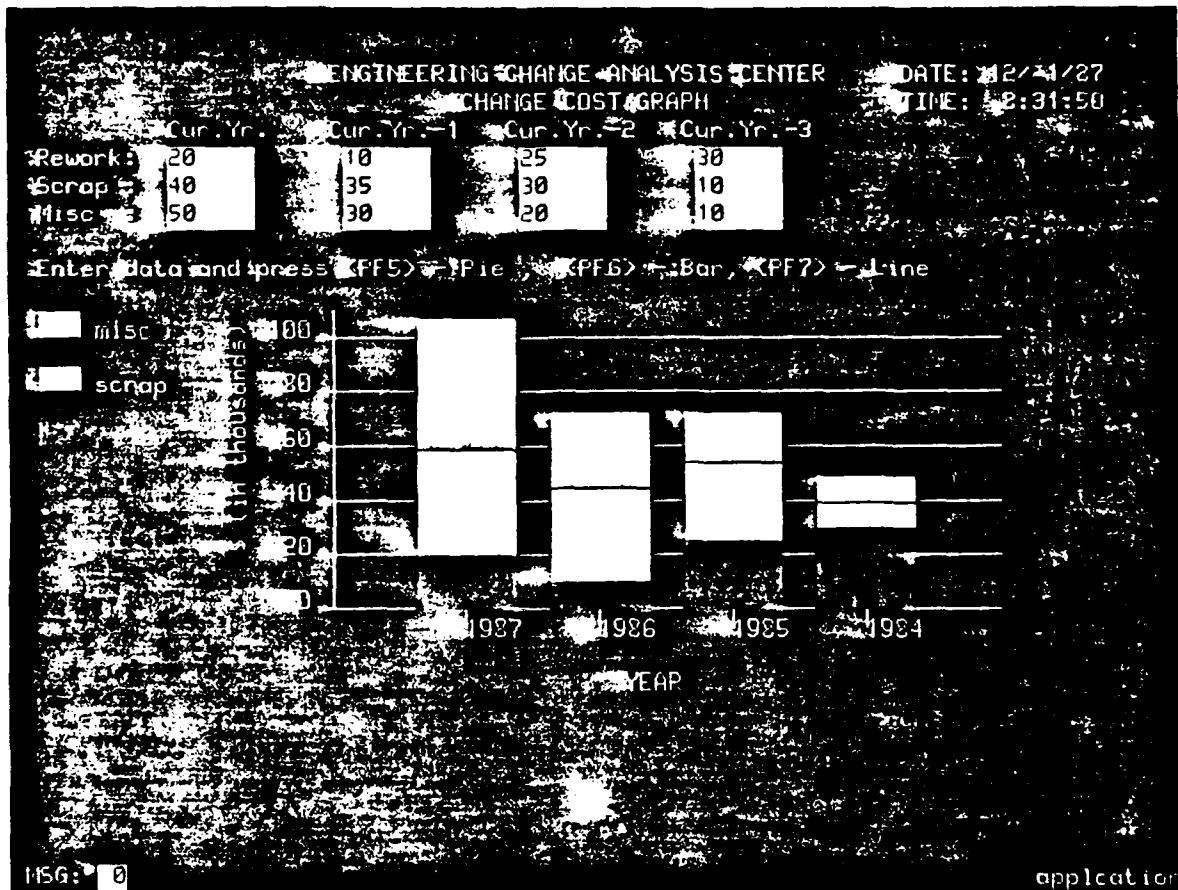
create pie graph piegraf
    at 3 15
    size 50 by 9
    using
('datafrm(1).rework','datafrm(1).scrap','datafrm(1).misc')
    attribute bluetext prompt (display blue)
    attribute whittext prompt (display white)

pie 1
    shade color red
    percent whittext outside

pie 2
    shade color white
    explode 20
    percent whittext outside

pie 3
    shade color blue
    percent whittext outside
```

Figure B-2 and corresponding GDL



```
create form grafrm2
  size 79 by 15
graph bargraf
  at 1 1
  size 79 by 15
  display as XPARNT

create bar graph bargraf
  using (1,2,3,4 AXIS AX1)
  attribute whiteline line (display white)
  attribute redtext prompt (display red)
  attribute bluetext prompt (display blue)
  attribute whittext prompt (display white)
  legend at 2 2

curve rework
  'datafrm(*).rework' using axis ax2
  legend redtext "rework"
  shade color red
  absolute

curve scrap
  'datafrm(*).scrap' using axis ax2
  legend whittext "scrap"
  shade color white
  additive using curve rework

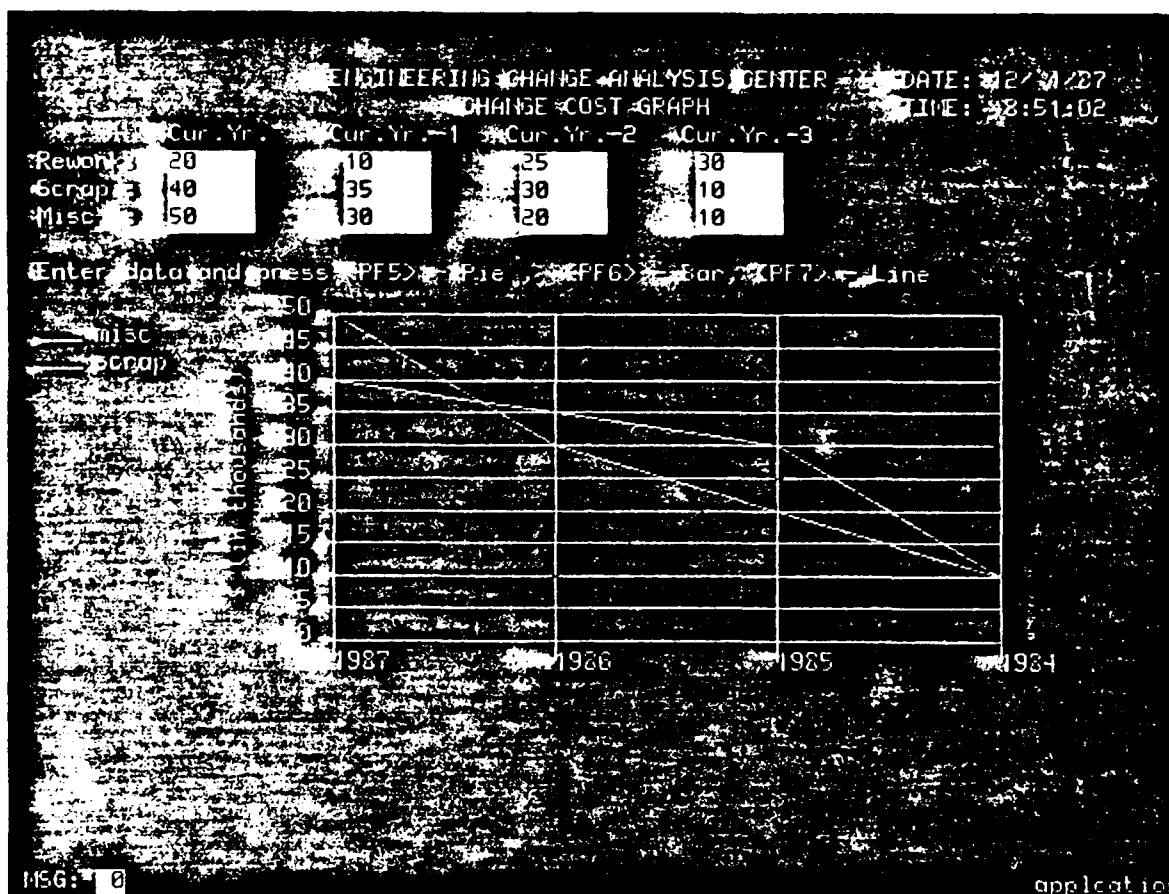
curve misc
  'datafrm(*).misc' using axis ax2
  legend bluetext "misc"
  shade color blue
  additive using curve scrap

axis ax1
  horizontal
  size 45
  display as whiteline
  at 12 22
  label whittext "YEAR"
  tick 4 whittext "1987" "1986" "1985" "1984"
```

UTP620344210
30 September 1990

```
axis ax2
  vertical
  size 12
  display as whiteline
  at 12 22
  min 0
  label whittext "    $ (in thousands)"
  grid
```

Figure B-3 and corresponding GDL




```
create form graffrm3
  size 79 by 15
graph linegraf
  at 1 1
  size 79 by 15
  display as XPARNT

create line graph linegraf
  using (1,2,3,4 AXIS AX1)
  attribute redline line (display red)
  attribute whiteline line (display white)
  attribute blueline line (display blue)
  attribute redtext prompt (display red)
  attribute whittext prompt (display white)
  attribute bluetext prompt (display blue)
  attribute cyanline line (display cyan)
  legend at 2 2

curve rework
  'datafrm(*).rework' using axis ax2
  legend redtext "rework"
  display as redline
  absolute

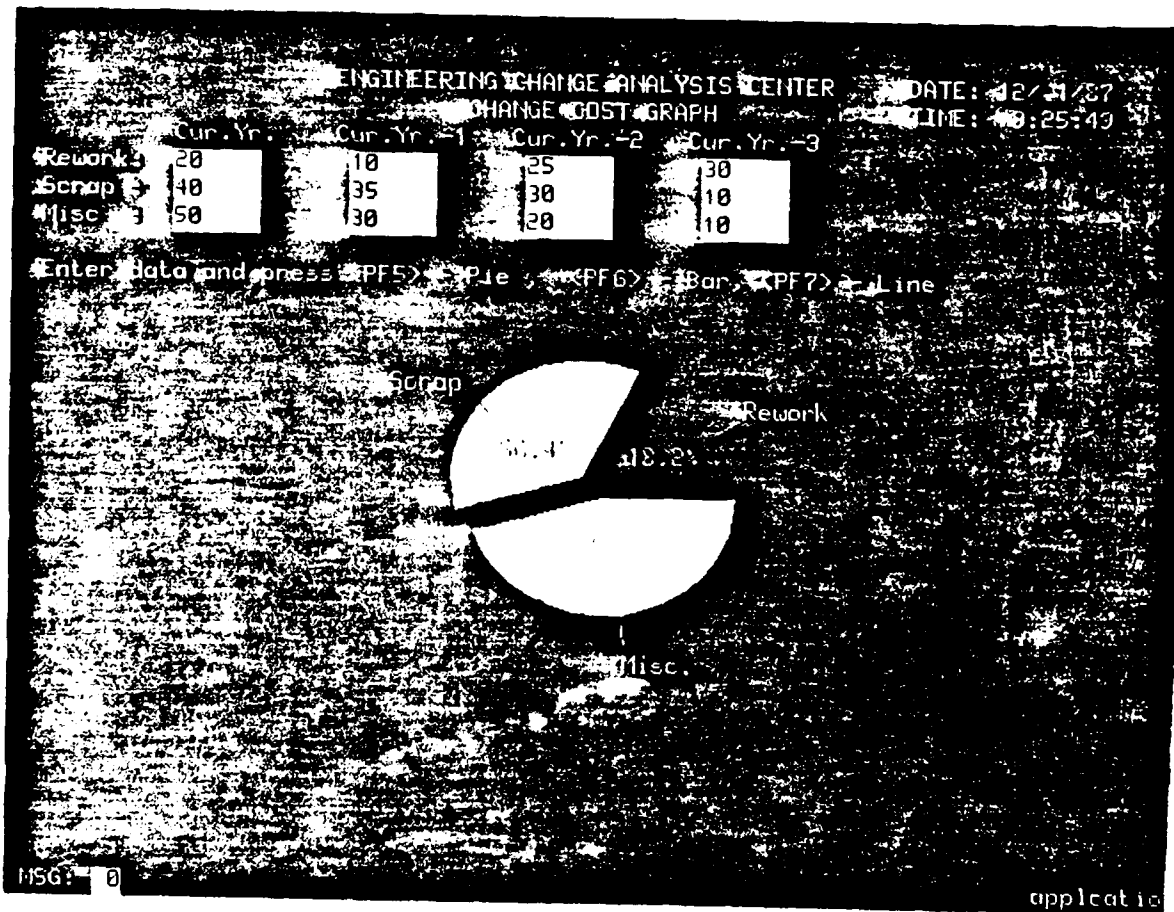
curve scrap
  'datafrm(*).scrap' using axis ax2
  legend whittext "scrap"
  display as whitline
  absolute

curve misc
  'datafrm(*).misc' using axis ax2
  legend bluetext "misc"
  display as blueline
  absolute

axis ax1
  horizontal
  display as cyanline
  at 13 22
  size 45
  label whittext "YEAR"
  tick 4 whittext "1987" "1986" "1985" "1984"
  grid
```

```
axis ax2
  vertical
  display as cyanline
  at 13 22
  min 0
  size 12
  label whittext "    $ (in thousands)"
  grid
```

Figure B-4 and corresponding GDL



```
create form graffrm4
    size 79 by 15
graph piegraf2
    at 1 1
    size 79 by 15
    display as XPARNT

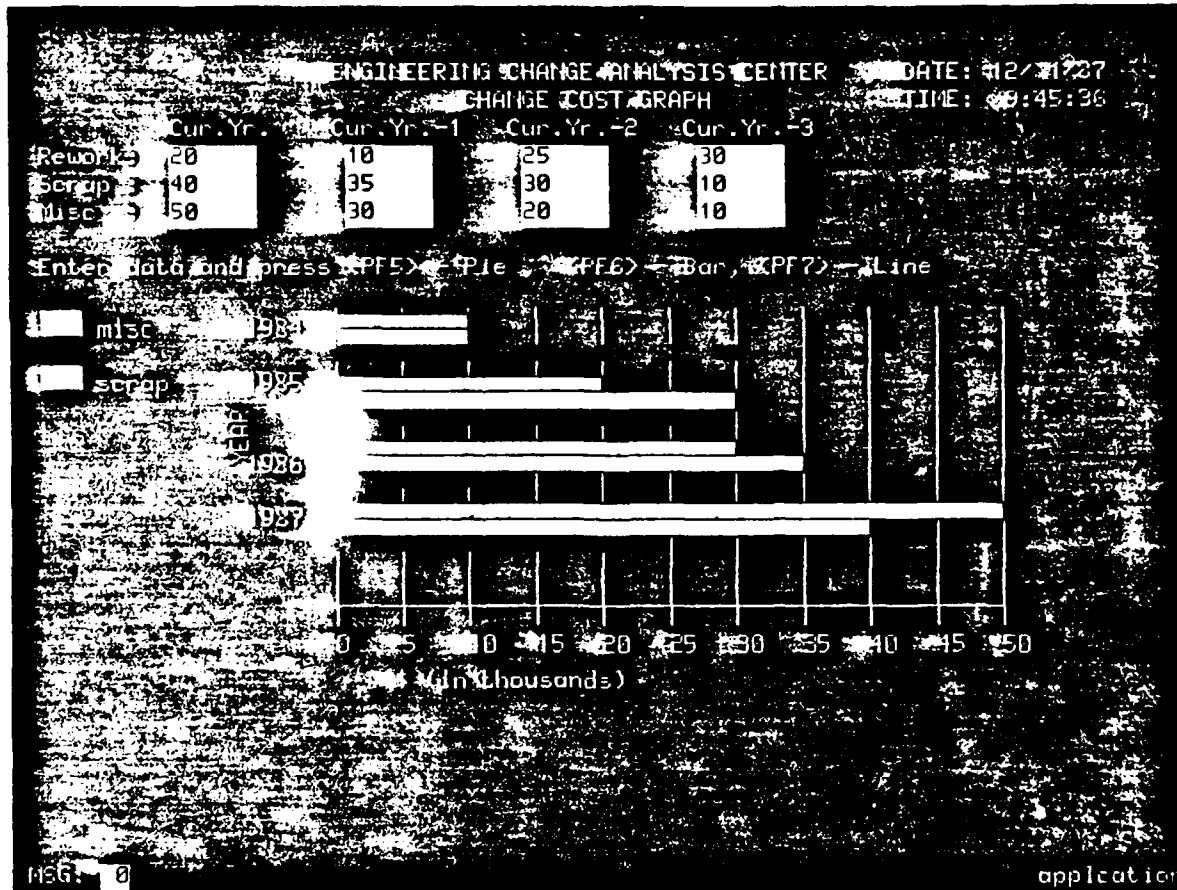
create pie graph piegraf2
    at 3 15
    size 50 by 9
    using
    ('datafrm(1).rework','datafrm(1).scrap','datafrm(1).misc')
    attribute bluetext prompt (display blue)
    attribute whittext prompt (display white)

pie 1
    shade color red
    percent whittext inside
    label whittext "Rework"

pie 2
    shade color white
    explode 20
    percent bluetext inside
    label whittext "Scrap"

pie 3
    shade color blue
    percent whittext inside
    label whittext "Misc."
```

Figure B-5 and corresponding GDL



```
create form graffrm5
  size 79 by 15
graph bargraf2
  at 1 1
  size 79 by 15
  display as XPARNT

create bar graph bargraf2
  using (1,2,3,4 AXIS AX1)
  attribute whiteline line (display white)
  attribute redtext prompt (display red)
  attribute bluetext prompt (display blue)
  attribute whittext prompt (display white)
  legend at 2 2

curve rework
  'datafrm(*).rework' using axis ax2
  legend redtext "rework"
  shade color red
  absolute

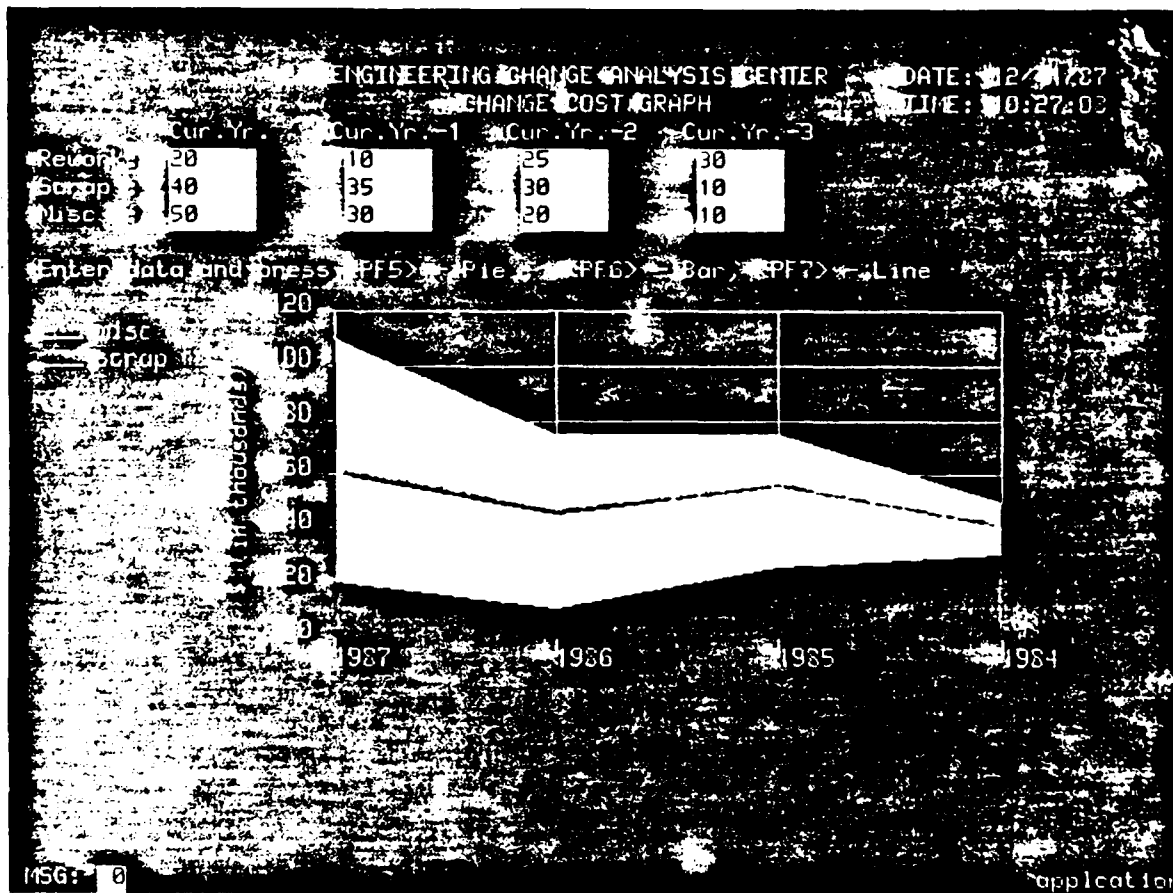
curve scrap
  'datafrm(*).scrap' using axis ax2
  legend whittext "scrap"
  shade color white
  absolute

curve misc
  'datafrm(*).misc' using axis ax2
  legend bluetext "misc"
  shade color blue
  absolute

axis ax1
  vertical
  size 12
  display as whiteline
  at 12 22
  label whittext "          YEAR"
  tick 4 whittext "1987" "1986" "1985" "1984"
```

```
axis ax2
  horizontal
  size 45
  display as whiteline
  at 12 22
  min 0
  label whittext "    $ (in thousands)"
  grid
```

Figure B-6 and corresponding GDL




```
create form graffrm6
  size 79 by 15
graph linegraf2
  at 1 1
  size 79 by 15
  display as XPARNT

create line graph linegraf2
  using (1,2,3,4 AXIS AX1)
  attribute redline line (display red)
  attribute whiteline line (display white)
  attribute blueline line (display blue)
  attribute redtext prompt (display red)
  attribute whittext prompt (display white)
  attribute bluetext prompt (display blue)
  attribute cyanline line (display cyan)
  legend at 2 2

curve rework
  'datafrm(*).rework' using axis ax2
  legend redtext "rework"
  display as redline
  absolute
  shade color red

curve scrap
  'datafrm(*).scrap' using axis ax2
  legend whittext "scrap"
  display as whiteline
  additive using curve rework
  shade color white

curve misc
  'datafrm(*).misc' using axis ax2
  legend bluetext "misc"
  display as blueline
  additive using curve scrap
  shade color blue

axis ax1
  horizontal
  display as cyanline
  at 13 22
  size 45
  label whittext "          YEAR"
  tick 4 whittext "1987" "1986" "1985" "1984"
  grid
```

```
axis ax2
  vertical
  display as cyanline
  at 13 22
  min 0
  size 12
  label whittext "    $ (in thousands)"
  grid
```

APPENDIX C

PRE-TEST PROCEDURES

This appendix describes the procedures for compiling and linking the test programs and for compiling the form definitions prior to running the tests.

For the first test, it is necessary to compile and link the test program. The following steps should be executed once the user has moved to the directory which contains the source code, grftst.c.

```
$ CC GRFTST  
$ DEFNTM  
$ @LGRFTST
```

Once the link has finished, the form definition file GRFTEST.FDL must be compiled. The following commands should be issued.

```
$ FLAN GRFTEST
```

The warnings may be ignored. All the necessary FD files will have been created and put in the location defined by IISSFLIB. NOTE that these steps may have been performed as part of the Configuration Management build process for the release.

For the second test, it is necessary to run the application generator. This may be performed in the IISS environment. Once an NTM is running, the IISS Login Screen (Figure 5-1) and Function Screen (Figure 5-2) may be brought up as described in Section 5. On the IISS Function Screen, enter APPGENER in the FUNCTION field. The following screen will appear when the <ENTER> key is pressed:

UIMS Application/Report Writer Generator

File Name:

Database/Password:

MSG: 0 application

Figure C-1 RAP Input Screen

Enter GRFDEMO onto the screen where the ADL file name is requested. Once the RAP has generated the FD files and the application, an application terminated message is displayed. Press the <QUIT> key to return to the IISS Function Screen. Press the <QUIT> key again to return to the system prompt.

Once the VAX system prompt is obtained, the program needs to be compiled and linked using the following commands:

```
$ CC GRAFDE
$ @cmdir:[ui]LNKAPC GRAFDE nomap GRAFDE
```

Before the test programs can be run, they must be defined in the UI database using SYSGEN. This is done as follows:

\$ VT100

Fill in the fields on the IISS
Logon Screen as follows:

Username: MORENC
Password: STANLEY
Role : MANAGER
Press <ENTER>

Fill in the FUNCTION field on the
IISS Function Screen as follows:

FUNCTION: SYSGEN
Press <ENTER>

The SYSGEN main menu screen will
be displayed

Press <PF7>

In the input field enter
"GRAFDE".

Press <PF7>

Enter the following information:

Description: Business Graphs Interactive Test Program
Name: SDGRAFDEZZ
Press <ENTER>

When the input field appears
under Authorized Roles, enter
"*".

Press <ENTER>

Application acknowledges entry.

Press <QUIT>

Displays the SYSGEN main menu.

Press <PF7>

In the input field enter
"GRFTST".

Press <PF>

Enter the following information:

Description: Business Graphs Test Program
Name: SDGRFTSTZZ
Press <ENTER>

When the input field appears
under Authorized Roles, enter
"*".

Press <ENTER>	Application acknowledges entry.
Press <QUIT>	Displays the SYSGEN main menu.
Press <QUIT>	Displays the IISS Function Screen.
Press <QUIT>	Returns to the system prompt.

The test program GRAFDE and GRFTST are now defined to IISS and the unit test may be performed.